Review of Approaches for Linked Data Ontology Enrichment

S. Subhashree, Rajeev Irny, and P. Sreenivasa Kumar(✉)

Department of Computer Science and Engineering,
Indian Institute of Technology - Madras, Chennai, India
{ssshree,rajeeviv,psk}@cse.iitm.ac.in

Abstract. Semantic Web technology has established a framework for creating a "web of data" where the nodes correspond to resources of interest in a domain and the edges correspond to logical statements that link these resources using binary relations of interest in the domain. The framework provides a standardized way of describing a domain of interest so that the description is machine-processable. This enables applications to share data and knowledge about entities in an unambiguous manner. Also, as all resources are represented using IRIs, a massive distributed network of datasets gets created. Applications can dynamically discover these datasets, access most recent data, interpret it using the associated meta-data (ontologies) and integrate them into their operations. While the Linked Open Data (LOD) initiative, based on the Semantic Web standards, has resulted in a huge web corpus of domain datasets, it is well-known that the majority of the statements in a dataset are of the type that link specific individuals to specific individuals (e.g. Paris is the capital of France) and there is major need to augment the datasets with statements that link higher-level entities (e.g. A statement about Countries and Cities such as "Every country has a city as its capital"). Adding statements of this kind is part of the task of enrichment of the LOD datasets called "ontology enrichment". In this paper, we review various recent research efforts that address this task. We investigate different types of ontology enrichments that are possible and summarize the research efforts in each category. We observe that while the initial rapid growth of LOD was contributed by techniques that converted structured data into the LOD space, the ontology enrichment is more involved and requires several techniques from natural language processing, machine learning and also methods that cleverly make use of the existing ontology statements to obtain new statements.

Keywords: Linked data · Knowledge enrichment · LOD enrichment
T-Box enrichment · Schema enrichment

1 Introduction

The Semantic Web (aka Web of data or Web 3.0) enables data from one source to be linked to any other source and to be "understood" by machines so that

© Springer International Publishing AG 2018
A. Negi et al. (Eds.): ICDCIT 2018, LNCS 10722, pp. 27–49, 2018.
https://doi.org/10.1007/978-3-319-72344-0_2

they can perform increasingly sophisticated tasks without human supervision. Semantic Web is often perceived as complementary to the World Wide Web, while it is actually an extension to the World Wide Web. It provides a framework to add new data and metadata to augment the existing web of documents. The Semantic Web technologies bring forth a new "web of data" paving the way for software agents to integrate data from diverse sources in a meaningful manner. RDF (Resource Description Framework) is the technology used to represent the nodes and edges of this new web of data. Linked Data is the particular realization of the web of data and it has now become a major constituent of the Semantic Web [1].

Linked Data refers to a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF. The LOD community project[1] works with the main objective of publishing open datasets as RDF triples and establishing RDF links between entities from different datasets. LOD complements the World Wide Web with a data space of entities connected to one another with labelled edges, which represent the relations among entity pairs (or entities and literal values).

With over 1014 interlinked datasets[2] across diverse domains such as life science, geography, politics, etc., the Linked Data initiative now supports a variety of applications ranging from semantic search to open domain question answering. For example, the Google's Knowledge Graph which is powered (partly) by the Freebase linked dataset is now being used by Google to enhance its search results with semantic-search information gathered from a wide variety of sources[3]. While many prominent organizations have started realising and exploiting the potential of linked datasets, these linked datasets are far from being complete [54]. More domains need to be covered, and more entities, concepts and links between them are required to be represented as RDF to enable improved and more intelligent usage of Linked Data. Sophisticated question answering systems like Watson which have linked datasets as part of their knowledge sources make use of the enriched linked datasets to answer more number and also a wider range of questions. The Linked Data community has realised the importance of enriching the linked datasets and hence the number of efforts towards enriching linked datasets in LOD have increased immensely in the past few years. A comprehensive study of the works done on Linked Data enrichment so far will help the community to understand the impact of LOD enrichment and its future scope.

1.1 Preliminaries

In this sub-section, we describe the important terms involved in the context of Semantic Web.

A *resource* is a real-world object we want to describe, and it is represented using an URI. A *class* (aka *concept* or *type*) is a group of resources, which is

[1] http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData.

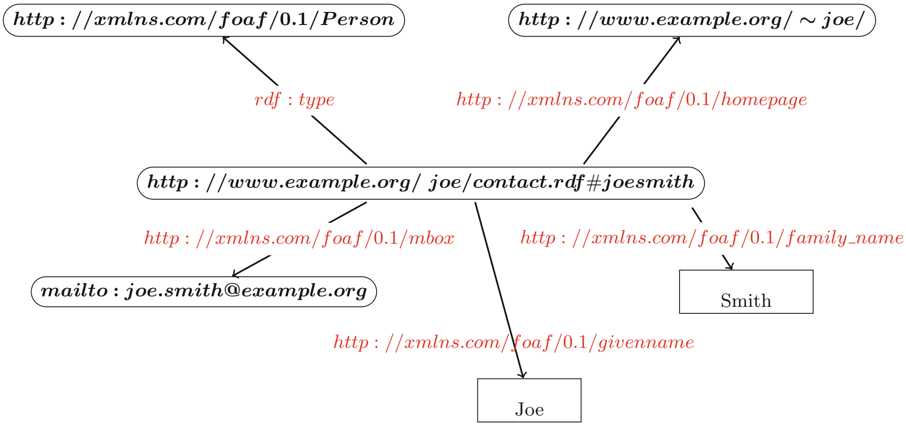
[2] http://lod-cloud.net/state/state_2014/.

[3] http://neilpatel.com/blog/the-beginners-guide-to-the-googles-knowledge-graph/.

also a resource by itself. A *property* (aka *role* or *relation* or binary *predicate*) is a relation between resources, and is also a resource by itself. A *statement* (aka an RDF *triple*) is composed of three parts - (*subject*, *predicate*, *object*) where the subject is a resource, predicate is a property and object is a resource or a *literal*. A literal is a constant value such as a string or a date. Given below is an example of an RDF triple:

(<http://dbpedia.org/resource/Barack_Obama>,
<http://dbpedia.org/ontology/birthPlace>,
<http://dbpedia.org/resource/Honolulu>).

A statement can be represented as a directed edge of a graph or as a triple or in XML (Fig. 1, Listings 1.1 and 1.2 respectively[4]).



http : //xmlns.com/foaf/0.1/Person
http : //www.example.org/ ∼ joe/
rdf : type
http : //xmlns.com/foaf/0.1/homepage
http : //www.example.org/ joe/contact.rdf#joesmith
http : //xmlns.com/foaf/0.1/mbox
http : //xmlns.com/foaf/0.1/family_name
mailto : joe.smith@example.org
Smith
http : //xmlns.com/foaf/0.1/givenname
Joe

Fig. 1. RDF graph representation

```
@prefix      : <http://www.example.org/~joe/contact.rdf#> .
@prefix foaf : <http://www.xmlns.com/foaf/0.1> .
@prefix rdf  : <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

:joesmith a foaf:Person .
       foaf:givenname"Joe";
       foaf:last_name"Smith";
       foaf:homepage <http://www.example.org/~joe/>;
       foaf:mbox <mailto:joe.smith@example.org> .
```

Listing 1.1. Triple Representation

[4] http://www.obitko.com/tutorials/ontologies-semantic-web/rdf-graph-and-syntax.html.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:foaf="http://www.xmlns.com/foaf/0.1"
    xmlns="http://www.example.org/~joe/contact.rdf#">

  <foaf:Person rdf:about=
    "http://www.example.org/~joe/contact.rdf#joesmith">
   <foaf:mbox rdf:resource="mailto:joe.smith@example.org"/>
   <foaf:homepage rdf:resource="http://www.example.org/~joe/"/>
   <foaf:family_name>Smith</foaf:family_name>
   <foaf:givenname>Joe</foaf:givenname>
  </foaf:Person>
</rdf:RDF>
```

Listing 1.2. RDF/XML Representation

There are two types of properties. An *object property* is a property between two resources while a *datatype property* is a property between a resource and a literal. The *domain* of a property is an assertion about the type of the resources that occur as subject of the property. Similarly, the *range* of a property is an assertion about the type of the resources that occur as object of the property. Domain (range) statements are also sometimes considered as domain (range) restrictions as they impose certain restrictions on the individuals that can be in the subject (object) position of a statement. For example, regarding the object property birthPlace mentioned above, one can state that its domain is a concept called Person and its range is Place. This allows us to infer that Barack_Obama is of type Person and Honolulu is of type Place.

An *ontology* is an explicit and formal representation of knowledge about a domain. It consists of classes, properties, axioms relating the classes and properties, and individuals of the domain. Statements in an ontology are divided into *T-Box* and *A-Box*. The T-Box is the terminological component of the ontology. It consists of class descriptions, properties and axioms involving them. The A-Box forms the assertion component of the ontology. Statements about the individuals (instances) fall into the A-Box.

Class expressions are used to give a detailed description about a class. There are two different types of class expressions:

- Atomic concept - denoted by a concept name Eg.: Person.
- Compound concept - denoted as a class expression involving one or more of the following operators (where A and B are class expressions, R is a property or role):
  1. Union of classes - $A \sqcup B$
     For example, the expression $Father \sqcup Mother$ can be used to describe the class $Parent$.

  2. Intersection of classes - $A \sqcap B$
     For example, the expression $Male \sqcap Parent$ can be used to denote the class $Father$.

  3. Complement of a class - $\neg A$
     For example, the expression $\neg Male$ can be used to describe the class of individuals who are not in the $Male$ class.

4. Existential restriction - $\exists R.A$
   This expression denotes the class of all individuals that are related to some individual of type A through a relation R. For example, the expression $\exists hasChild.Female$ can be used to describe the class of individuals who have daughters.

5. Universal restriction - $\forall R.A$
   This expression denotes the set of all those individuals whose all R-successors belong to the class A (if (x,R,y) is a triple, y is called an R-successor of x). For example, the expression $\forall hasChild.Female$ can be used to describe the class of individuals who have only daughters as children.

6. Cardinality restriction - $\leq nR.A$
   This expression denotes the set of all individuals that have at most n R-successors. Similarly, $\geq nR.A$ can be used to place a lower bound on the R-successors. For example, the expression $\geq 2hasChild.Female$ can be used to describe the class of individuals who have at least 2 daughters.

Different Description Logics are formed from subsets of these operators, more details of which can be found in [3]. The above mentioned description logic (DL) notation is used occasionally in the paper to give class descriptions.

Important ontology frameworks and languages are listed below:

- RDF - Resource Description Framework - defines constructs which are the building blocks of the Semantic Web such as classes and properties. E.g.: rdf:type
- RDFS - Resource Description Framework Schema - defines properties and classes of RDF resources. E.g.: rdfs:subClassOf
- OWL - Web Ontology Language - defines richer ontology constructs. E.g.: owl:disjointWith, owl:sameAs
- SPARQL - SPARQL Protocol and RDF Query Language - a query language similar to SQL in syntax. It is used to query the triples in linked datasets.

1.2 Prominent Linked Data Projects

DBpedia: DBpedia [32] is one of the most popular linked datasets and has been developed based on a crowd-sourced community effort. DBpedia is composed of the structured information extracted from Wikipedia articles and is represented in triple format. Currently, DBpedia is available in 125 languages. The English version of the DBpedia Knowledge Base (KB) currently describes 6.6 million entities, out of which, 5.5 million are described in a consistent ontology[5] including

[5] http://wiki.dbpedia.org/downloads-2016-10#dbpedia-ontology.

1.5 million persons, 840 K places, 496 K works, 286 K organizations, 306 K species, 58 K plants and 6 K diseases[6].

YAGO: YAGO is a huge linked dataset constructed through automatic extraction from sources such as Wikipedia and WordNet. The current version of YAGO, namely, YAGO3 [34] has around 10 million entities (of types persons, organizations, cities, etc.) and contains more than 120 million facts about these entities[7].

LinkedMDB: LinkedMDB is the first open Semantic Web dataset for movies. It contains links to other datasets such as DBpedia, Geonames, etc. and to websites such as IMDb. A few important classes of LinkedMDB include films, actors, movie characters, directors, producers, editors, writers, music composers, soundtracks, and movie ratings[8].

1.3 Categories of LOD Enrichment

This section categorizes and describes the different ways in which the linked datasets in LOD can be enriched. We can classify LOD enrichment works broadly into two types: T-Box enrichment and A-Box enrichment. T-Box enrichment includes the following: discovering property axioms, discovering class axioms, discovering new properties, and discovering new classes. A-Box enrichment involves the following: discovering owl:sameAs links[9], discovering instances of a class (type assertions), discovering instances of existing relations, detecting erroneous type assertions, detecting erroneous relations, detecting erroneous literal values, and detecting erroneous owl:sameAs links.

The focus of this survey is to provide a comprehensive overview of the works proposed for T-Box enrichment. A recent study on knowledge graph[10] refinement approaches [41] can be referred to for works on A-Box enrichment. It should be noted that a knowledge graph mainly consists of individual members (of classes) and relations among them [41] - i.e. a knowledge graph focusses on its A-Box while its T-Box plays a minimal role. However, in the context of Linked Data, the goal is to add more semantics to the dataset which is possible only when we enrich the T-Box (schema) of the linked dataset. As this paper is written from the perspective of LOD enrichment rather than Knowledge Graph enrichment, we mainly focus on T-Box enrichment techniques. However, if there are Knowledge

[6] http://wiki.dbpedia.org/datasets/dbpedia-version-2016-10.

[7] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/.

[8] http://www.linkedmdb.org/.

[9] owl:sameAs is a built-in OWL property which links an individual to another individual denoting that the two resources represent the same real-world entity.

[10] The term Knowledge Graph was coined by Google in 2012, referring to their use of semantic knowledge in Web Search. The term is recently being used in a broader sense: any graph-based representation of some knowledge could be considered a knowledge graph.

Graph enrichment techniques which also focus on T-Box enrichment, we include them in this survey. Papers which discuss ontology building (from the ground-up) are not dealt with in this survey as they are not enrichment works i.e. in such papers, a partially built ontology does not exist.

The rest of the paper is organized as follows: Sect. 2 gives an account of the techniques proposed in the literature for discovering property axioms. A brief summary of approaches proposed in the literature for discovering class axioms is given in Sect. 3. Sections 4 and 5 describe the systems proposed for discovering new properties and new classes respectively. Conclusions drawn from the survey are given in Sect. 6.

2 Discovering Property Axioms

Properties in Linked Data provide semantic associations between instances in Linked Data and thus are indispensable in representing information in the semantic web. Property Axioms give additional information about predicates and the various Property Axioms that can be used are listed in Table 1. Additional details on the semantics of these axioms can be found in [3]. Most linked datasets in the LOD are deficient in Property Axioms. Thus, a considerable effort has been directed towards enriching schemas associated with datasets in LOD by discovering Property Axioms. We categorize these methods to discover axioms as : (1) Instance based and (2) Schema based methods. The instance based methods rely upon triples in the linked dataset while the schema based methods utilize the schema information like Domain or Range restrictions, type statements to enrich the linked datasets with axioms. We discuss these methods in detail below:

Table 1. Semantics of Property Axioms: r_i, r_j are properties in a linked dataset KB and x, y, z are distinct instances in the KB. Here, $r_i(x, y)$ denotes a triple $< x\, r_i\, y >$ in the KB.

Axiom	Semantics
Subsumption (r_i, r_j)	$r_i(x,y) \in \text{KB} \implies r_j(x,y) \in \text{KB}$
Equivalence (r_i, r_j)	$r_i(x,y) \in \text{KB} \implies r_j(x,y) \in \text{KB} \wedge r_j(x,y) \in \text{KB} \implies r_i(x,y) \in \text{KB}$
Symmetry (r_i)	$r_i(x,y) \in \text{KB} \implies r_i(y,x) \in \text{KB}$
Inverse (r_i, r_j)	$r_i(x,y) \in \text{KB} \implies r_j(y,x) \in \text{KB} \wedge r_j(x,y) \in \text{KB} \implies r_i(y,x) \in \text{KB}$
Asymmetry (r_i)	$r_i(x,y) \in \text{KB} \implies r_i(y,x) \notin \text{KB}$
Transitivity (r_i)	$r_i(x,y) \in \text{KB} \wedge r_i(y,z) \in \text{KB} \implies r_i(x,z) \in \text{KB}$
Disjoint (r_i, r_j)	$r_i(x,y) \in \text{KB} \implies r_j(x,y) \notin \text{KB}$
Functionality (r_i)	$r_i(x,y) \in \text{KB} \wedge r_i(x,z) \in \text{KB} \implies y = z$
Inverse functionality (r_i)	$r_i(x,y) \in \text{KB} \wedge r_i(z,y) \in \text{KB} \implies x = z$

2.1 Instance Based Methods

Methods under this category leverage the large number of triples present in the linked datasets and use statistical techniques like classification, clustering and association rule mining to discover axioms. Fleischhacker et al. [19] proposed a method to discover all the property axioms shown in Table 1. The method involved the use of an off-the-shelf association rule miner [9] to mine association rules. The inputs to this rule miner were transaction tables which are created from the linked datasets. Each row in the transaction tables represents a pair of instances in the linked data and all the predicates that hold between them. The rules mined by the rule miner are checked against a set of predefined patterns. Rules matching these predefined patterns are selected to be converted to property axioms. Most other works in the literature concentrate on discovering subsumption and equivalence axioms.

Galárraga et al. [22] proposed Association Mining under Incomplete Evidence (AMIE) for mining closed Horn rules under incomplete evidence. The rules generated by AMIE are of the form shown in Eq. (1). Here r is a predicate in the linked dataset and B_i is a triple of the form $<?x\ p_i\ ?y>$ and x, y are placeholder variables for instances in the linked dataset. As such $\overrightarrow{B}$ is called the *Body* of a rule and $r(x, y)$ is called the head of the rule. A rule produced by AMIE is closed, i.e. every variable is in the rule occurs in multiples of two and always in pairs.

$$\begin{aligned} \overrightarrow{B} &\Rightarrow r(x, y) \\ \overrightarrow{B} &= B_1 \wedge B_2 \wedge .... \wedge B_n \end{aligned} \tag{1}$$

The Horn rules in Eq. (1) represent the correlations between properties in the dataset. To ensure efficient computation of the Horn rules, Galárraga et al. proposed several logical constraints in the form of refinement operators. Galárraga et al. [22] also proposed the notion of PCA (Partial Completeness Assumption) which makes concessions in selecting negative assertions for a rule. A negative assertion for the Horn rule shown in Eq. (1) is a subject object pair (x, y) such that it is a valid instantiation of the *Body* of the rule but is an invalid instantiation of the *Head* of the rule. PCA states that for a rule shown in Eq. (1), given a predicate r and its subject x, if we know one corresponding object y then we may assume that we know all objects y- that is the objects associated with x in the data are the only ones that x can get associated with.

In [21], the Horn rules generated by AMIE are interpreted as subsumption or equivalence axioms. The interpretation of rules is based on a set of patterns called ROSA (Rule for Ontology Schema Alignment) rules, shown in Fig. 2. Each rule that fits the patterns shown in Fig. 2 is also associated with a PCA confidence score. The same technique can also be used to align equivalent predicates across heterogeneous datasets. This involves first aligning the instances in the two datasets using the owl:sameAs links between instances in the two datasets prior to mining the rules. Once we have all the rules generated by AMIE, all it takes to identify equivalent predicates across datasets is to compare the rules against

$$\begin{aligned} &r(x,y) \Rightarrow r'(x,y) && \text{(Property Subsumption)} \\ &r(x,y) \iff r'(x,y) && \text{(Property Equivalence)} \end{aligned}$$

Fig. 2. ROSA rules for Property Subsumption and Equivalence

the ROSA rule for Property Equivalence. A owl:sameAs link between two entities in Linked Data signifies that they are synonymous. For instance, the instances yago:Barack_Obama and dbr[11]:Barack_Obama are resources to identify the same person and hence will be linked by a owl:sameAs link. For datasets where the owl:sameAs links are not mentioned, Galárraga et al. in [20] introduce a technique to canonicalize the instances and predicates across heterogeneous linked datasets. They do this by first clustering synonymous instances using a technique called Token Blocking [40]. Under such a technique synonymous instances in the two datasets like President_Obama and Barack_Obama will be placed in the same cluster implying that they refer to the same entity. Post this clustering, we obtain a normalized dataset where instances have been aligned. Using these aligned instances, we can now align equivalent predicates across the two datasets. To obtain equivalent predicates a similar procedure involving use of AMIE and ROSA rules can be employed. Each of the equivalent predicate pairs discovered in the above methods can be added to the ontology as Equivalent Property Axioms. The same holds true for the Subsumption Property Axioms.

On similar lines, in our recent work [25], we discover latent Inverse and Symmetric axioms in linked datasets. In this work we outline the challenges involved in discovering latent property axioms by an instance based method and then propose measures to overcome these challenges. One such challenge is the presence of synonymous predicates in the Linked Data and a higher preference to use one of them. For instance, we have dbo:infuenced and dbo:influencedBy as predicates in DBpedia, these predicates convey similar meaning but are inverse of each other. However, dbo:influencedBy is more frequently used among the two to make an assertion. This points to a preference of one predicate over the other which makes the discovery of inverse axioms a challenging task. To this end, we introduced predicate-preference factor (*ppf*) to account for the difference in frequency of use of synonymous (but inverse) predicates. Also, to remedy the lack of reliable and useful domain and range information in linked datasets, we introduced a novel semantic-similarity measure which uses the rdf:type information of instances in the subject and object of a predicate to suggest the reliable axioms. Through experiments we show that the proposed method discovers twice as many axioms, at improved accuracy.

The triples in a linked dataset can also be visualized as a graph with the instances in the subject/object of a triple as nodes and the predicates as edges. Based on this view of linked dataset as a graph, many works apply graph mining techniques to extract meaningful semantic associations between the nodes or edges in the graph. However, most of these approaches consider just the instance-level

[11] http://dbpedia.org/resource.

information (i.e. triples) to suggest rdf:type statements [8,27], to summarize graph entities [47] or query re-writing [55].

2.2 Schema Based Methods

Methods in this section use the schema-level knowledge in addition to using the instance-level information. Axioms discovered by instance based methods do not use the schema information associated with the linked datasets. Work by Barati et al. [5] describe how the lack of schema could negatively impact the induction of axioms. To this end, they propose SWARM (Semantic Web Association Rule Mining), which generalizes Association Rule mining for the semantic web setting. SWARM adds semantics to the association rules by using schema-level knowledge such as rdf:type and rdfs:subClassOf statements. Augmenting the association rules with semantics allows us to interpret them as Behavioral Patterns. For instance, consider a rule mined by SWARM as shown below:

$$\{Person\} : (livesIn, Delhi) \Rightarrow (Speaks, Hindi)$$

The rule above means that the dataset contains many instances to support the pattern that a Person who is a resident of Delhi, speaks Hindi and SWARM uses such rules to identify behavioral patterns from the linked datasets.

Ontology Matching and Alignment techniques [43] involve finding correspondences among the properties either in the same ontology or across different ontologies. Recent advances in this field have given emphasis on the use of large linked datasets to align ontologies or match equivalent properties across ontologies based on the evidence in the linked datasets. For instance, Suchanek et al. propose PARIS [45], which automates the matching of instances, classes and properties across ontologies. PARIS presents a probabilistic approach to estimate the degree of overlap between the instances of two properties in the datasets under consideration. It processes the instances in the linked dataset as well as the ontologies associated with them to align equivalent predicates across datasets. To work with heterogeneous datasets, Suchanek et al. begin by finding equivalent instances across these datasets. They propose a probabilistic model to find equivalent instance pairs. For example, two instances $x \equiv x'$ holds if there is a common predicate r such that triples $r(x, y)$ and $r(x', y')$ exist in the datasets, $y \equiv y'$ and r is inverse-functional. Here x, y belong to one dataset while x', y' belong to another dataset. Observe that to align equivalent instances using the above method, a common predicate r must exist in the two datasets. In addition to finding equivalent instances, PARIS also attempts to discover equivalent predicates (r, r') across the two datasets and does so by using the instances aligned in the method mentioned above. To discover equivalent predicate pairs it checks for the existence of subsumption relation between them, i.e. $r \equiv r'$ if $r \sqsubseteq r'$ and $r' \sqsubseteq r$. Here, $r \equiv r'$ implies that r, r' are equivalent predicates and $r' \sqsubseteq r$ implies that r' is a sub-property of r. PARIS determines the probability that r' is sub-property of r i.e. $Pr(r' \sqsubseteq r)$ as the ratio of number of instance-pairs x, y in r' that are also in r. Note that with the discovery of new equivalent predicate pairs, we can update the probability of equivalence of two instances in the dataset which in turn updates the probability of equivalence of predicate pairs. Thus, the two steps of finding equivalent instances and equivalent predicates are iterated repeatedly until convergence, i.e. when the probabilities do not change any more. It is found experimentally that the convergence is reached after a few iterations. Details about how the probability values were calculated are explained in [45].

On similar lines, Koutraki et al. [28] propose *SORAL* (Supervised Ontology Relation Alignment), a supervised approach to learn the subsumption and equivalence property axioms. They propose the use of several ILP (Inductive Logic Programming) and frequency based features to model a binary classifier to determine if a pair of predicates form an equivalence or subsumption axiom. Some of these features are discussed below:

1. **ILP based features**: This set of features include the confidence measure calculated normally and confidence calculated under the partial completeness assumption [22]. PCA works best when predicates are functional or quasi-functional (The authors in [22] quantize the functionality of a predicate as a value between $0, 1$ where a value of 1 implies the predicate is functional and 0 otherwise. Quasi-functional predicates are those which have a functionality values close to 1). To overcome this drawback, Koutaki et al. introduce PIA (Partial Incompleteness Assumption) which can be considered as a weighted PCA for less functional predicates.
2. **Frequency based features**: These features consider statistics of entities in the dataset like cardinality of relations, type distributions of predicates etc. The features under this category include the functionality of predicates, Jaccard Similarity between the type distributions of 2 predicates. These features also include joint probabilities of confidence score calculated normally and under PCA.

It is worth noting that the training data used in the learning algorithm was created by the authors. Thus, being a supervised technique to align predicates, it is dependent on existence of a training resource. Koutraki et al. [28] also suggest a method to alleviate the challenges of handling large linked datasets by using some sampling techniques. They present experimental results for sample size 100, 500 and 1000. Through experiments Koutraki et al. show that a supervised method to learn subsumption and equivalence axioms based on degree of overlap of instance between two relations is effective in matching properties across ontologies.

However, a major drawback of techniques described above is assuming the existence of common instances between ontologies. While it is a reasonable assumption to make, the methods that depend on common instances fail when the ontologies being aligned share very few or no common instances. To overcome the lack of common instances in an ontology, Wijiya et al. [52] propose PIDGIN, a system that supplements the lack of common overlapping instances between ontologies with the information present in large natural language corpus. They use the corpus to ground the relations and instances in the ontology to verbs and instances in the corpus respectively. This makes up for the lack of common instances between ontologies being matched.

Domain and Range Restrictions. Often overlooked albeit important part of ontologies are the Domain and Range restrictions related to properties. These restrictions ensure that the instances in the subject or object of a property are of the correct class-type. For instance dbo:manager has class dbo:SportsTeam as Domain and class dbo:Person as Range, which means that the instances in the subject of dbo:manager should belong to the class dbo:SportsTeam. However, Tonon et al. [48] show that in most linked datasets, the domain and range restrictions are violated. Thus, we can enrich the corresponding ontology by updating the domain and range restrictions based on the evidence in the linked datasets. For example, consider the property dbo:manager above. Even though the DBpedia ontology mentions dbo:SportsTeam, the instances in the linked data suggests that the domain should be dbo:SportsSeason.

Work by Tonon et al. [48] explores determining the domain and range of properties based on the instances in the linked dataset. They propose LeXt and ReXt to suggest the instance based domain and range of properties. The LeXt performs a depth-first search on the class-type hierarchy for each instance in the subject of a property to statistically determine the most specific class of instances occurring as the domain of the properties. Similarly ReXt determines the instance-based range of a predicate. Töpper et al. [49] also propose a frequency based method to suggest the domain and range of properties in linked dataset based on the class-types of the instances in the subject and object of a property.

3 Discovering Concept Axioms

In ontologies, Concept Axioms play an important role in expressing the relationships that hold between the different Concepts. The semantics of Concept axioms are shown in Table 2 where we see that compared to property axioms, class axioms are less diverse.

Töpper et al. [49] motivated the need for disjoint axioms as a means to find inconsistencies in a linked dataset. They propose to find similarity between two concepts in the ontology, thus, those concept pairs that have similarity scores below a fixed threshold are considered disjoint. To this end, they represent a concept (C) in the vector space. The length of the vector is equal to the number of properties

Table 2. Semantics of Concept Axioms: C_i, C_j are concepts in a linked dataset KB and x, y, z are distinct instances in the KB. Here, $C_i(x)$ denotes that x is of class-type C_i in the KB.

Axiom	Semantics
Subsumption (C_i, C_j)	$C_i(x) \in \text{KB} \implies C_j(x) \in \text{KB}$
Equivalence (C_i, C_j)	$C_i(x) \in \text{KB} \implies C_j(x) \in \text{KB} \wedge C_j(x) \in \text{KB} \implies C_i(x) \in \text{KB}$
Disjoint (C_i, C_j)	$C_i(x) \in \text{KB} \implies C_j(x) \notin \text{KB}$

in the dataset. The weight of each property is modeled after *tf, idf* in Information Retrieval where the *tf* part denotes frequency of occurrence of the property with class C in the dataset and the *idf* part denotes the general relevance of the property in the dataset. Additionally, Fleischhacker et al. [18] propose a method to inductively learn disjointness axioms. They discuss multiple strategies like learning correlation between two concepts based on the count of common instantiations between them. Thus, concepts that have very low or negative correlation are considered to be disjoint with each other. Another technique they suggest is similar to [19] where the difference lies in the representation of rows (discussed in Sect. 2.1). In this case, a row in the transaction table represents the set of concepts that an instance belongs to.

Similar to property axioms, a large portion of the work in literature discusses the discovery of concept subsumption axioms. The set of all subsumption axioms in an ontology aid in creating the class hierarchy or the taxonomy while equivalence axioms are mostly used to align two different ontologies. Volker et al. [51] propose a framework which is a precursor to [19], explained in Sect. 2.1. As explained in the discussion about the disjointness axioms above, the difference between [19,51] is in the representation of transaction tables and in the patterns that are matched to interpret association rules as axioms. Li et al. [33] suggest an improvement over [51] by proposing a method to mine axioms more efficiently. It involves dividing the linked dataset into several blocks (based on disjoint properties) to facilitate the application of mining axioms in parallel. Also note that the methods [21,45] mentioned in Sect. 2 can also be used to find equivalent and subsumption class axioms.

In addition to the axioms shown in Table 2, [33,51] also discover class expressions like $C_i \sqsubseteq \exists r.C_j$ or $\exists r.C_j \sqsubseteq C_i$. Here C_i, C_j are concepts and r is a property in an ontology. The class expression above can be considered as a specialized form of subsumption axioms where the latter expression suggests that whenever we have a triple $< x\ r\ y >$ in KB and $C_j(y)$, then we have $C_i(x)$. Such class expression are useful in describing class definitions. For instance for the expression $C_i \sqsubseteq \exists r.C_j$, if r is *authorOf*, C_j is *Journal_Article* and C_i is *Doctoral_Advisor* then, it means that every *Doctoral_Advisor* besides other things has authored a *Journal_Article.*

The DL-Learner framework [12] encompasses various algorithms for inductive learning of concept axioms and class expressions. The procedure followed by the framework to detect axioms is as follows [11]: Frequent axiom patterns in various ontologies are discovered and converted into corresponding SPARQL query patterns. The query patterns are then applied to other datasets to enrich them with new axioms. For example, in the experiments conducted by [11], patterns have been mined from more than one thousand ontologies and then applied on the DBpedia dataset. A few patterns which were obtained among the top 15 patterns are given below:

A **SubClassOf** p **some** (q **some** B), or equivalently $A \sqsubseteq \exists p.(\exists q.B)$ in DL
A **equivalentTo** B **and** p **some** C, or equivalently $A \equiv B \sqcap \exists p.C$
A **SubClassOf** p **value** A, or equivalently $A \sqsubseteq \exists p.\{A\}$

A few axioms which were obtained by applying the above patterns on DBpedia are:

> Song **equivalentTo** MusicalWork **and** (artist **some** Agent) **and** (writer **some** Artist), or equivalently $Song \equiv MusicalWork \sqcap (\exists artist.Agent) \sqcap (\exists writer.Artist)$
> Conifer **SubClassOf** order **value** Pinales, or equivalently $Conifer \sqsubseteq \exists order.\{Pinales\}$

The algorithms proposed under the DL-Learner framework for learning of class expressions are described in Sect. 5.

3.1 Discussion

It is worth noting that most of the methods we discussed in this and in the previous section focus on the discovery of subsumption and equivalence axioms, be it property or concept axioms. These axioms, while crucial to formation of a class/property hierarchy, limit the diversity of the axioms in the ontology. We believe that expanding the scope of these methods to discover additional axioms namely, Functionality, Inverse functionality, Inverse, Transitivity will enhance the understanding of the underlying domain and also help in keeping the dataset consistent with the world-knowledge. Thus, the discovery of axioms that add value to the ontology is one of the promising areas of research. Additionally, with the use of PCA (and PIA), several works described in the Sections above compensate for the incomplete nature of data in the semantic web. While this is a step in the correct direction, a technique that is not restricted by the functionality of the predicates (like PCA) will surely provide a more versatile method to overcome the incompleteness in semantic web and thus is a potential future extension.

4 Discovering New Properties

Most of the linked datasets are deficient in the number of object properties they have. For example, the linked dataset YAGO has 488,469 classes [34]. Among such a huge number of classes, surprisingly there are only 32 object properties[12] and hence looking for more object properties to connect these classes becomes a necessary step towards enriching linked datasets. Details of the methods proposed in literature to add new object properties are given below:

Several works have been proposed to discover new object properties in the context of enriching the NELL (Never Ending Language Learner) Knowledge Base. NELL [37] is a part of the "Read the Web" project[13] which is an initiative

[12] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/statistics/ - totally there are 60 object properties, but 28 of them connect the domain class to the class http://dbpedia.org/class/yago/YagoLiteral.

[13] http://rtw.ml.cmu.edu/rtw/.

to create a machine that learns to read the entire web. NELL has been running continuously since January 2010 and it performs two main tasks - extract facts from web pages, and improve its learning techniques to extract more accurate facts in future. NELL has a few helper systems which aid it in extending its T-Box so that more instances of the newly discovered relations and concepts can be fetched by NELL from the web. Mohamed et al. proposed OntExt (Ontology Extension system) [38] which discovers new relations given two categories from the NELL ontology (classes are called as categories in the NELL KB). OntExt does this by extracting text patterns from the web corpus and clustering them based on co-occurrence values. For example, if the phrases "Ganges flows through Allahabad" and "Ganges in the heart of Allahabad" occur in the web corpus with a very high frequency then this is taken as an indicator that the patterns, "flows through" and "in the heart of" are similar to each other. When such an evidence is shown by many number of subject-object pairs, OntExt gives a very high similarity score between the two patterns. In general, OntExt works in the following manner: given a pair of categories and a set of sentences-each containing a pair of instances known to belong to the given categories, OntExt collects the words in between the instances from each sentence and calls these words a "context-pattern". Then it builds a co-occurrence matrix (context-pattern X context-pattern) which is based on the frequencies of occurrence of the context-patterns with the same subject-object instance pairs. For example, in the above case of finding relations between Rivers and Cities, if the pair "Ganges" and "Allahabad" occurs with the context-pattern "flows through" with a frequency f_1 and the pair occurs with the pattern "in the heart of" with a frequency f_2, then the matrix entry corresponding to these two context-patterns will be given a value of $(f_1 + f_2)$. In case there is another subject-object pair (for example-Thames, London) occurring with both these context-patterns with frequencies f_3 and f_4 respectively, then the matrix cell value becomes $(f_1 + f_2 + f_3 + f_4)$. K-means clustering is applied on the normalized matrix to group the related context-patterns together. The centroid of each cluster is proposed as a new relation. OntExt also generates the instances (subject-object pairs) of these new relations based on how often each subject-object pair co-occurs with a new relation in the web corpus. OntExt was followed by newOntExt [6,7] whose primary goal was to overcome certain challenges faced by OntExt and to make the ontology extension process scalable and feasible so that it can be effectively utilised on the NELL Knowledge Base. The authors incorporated the following changes in newOntExt: Instead of considering all the words in between the two input instances as a pattern, newOntExt used ReVerb [17] for extracting the patterns in order to reduce the number of noisy patterns obtained. In order to reduce the computational cost, a more elegant file structure was used for searching through the sentences. Instead of considering every pair of categories as input to this system, reduced category groups of interest were formed to pick the input category pairs (for example, the categories related to the domain of sports: sports league, sport, athlete and sports team). Later, Cergani et al. [13] identified the following issues in the clustering phase of newOntExt: An entity pair cannot be connected by multiple relations and even the most obvious outliers (noisy relations) cannot be removed by their clustering phase. Also, the number of clusters had to be specified before-hand. Hence they have proposed a minor improvement to overcome these issues by replacing the clustering phase of newOntExt with matrix factorization techniques such as Non-negative Matrix Factorization (NMF) and Boolean Matrix Factorization (BMF). The authors of [44] made the following observations w.r.t. the working of newOntExt: The number of incorrect relations produced by newOntExt is high mostly because the new relations are not filtered based on any contextual check. Even semantically dissimilar but meaningful relations are placed in the same cluster and hence important relations get dropped by newOntExt. Also, the relations produced are not grounded to the knowledge base (by grounding, we mean mapping of discovered relations to existing LOD object properties). In [44], the authors propose a system called DART (**D**etecting **A**rbitrary **R**elations for enriching **T**-Boxes of Linked Data) to enrich linked datasets with new object properties between two given classes by means of contextual similarity detection and paraphrase detection tools. DART performs grounding of relations and is also shown to be better than newOntExt in terms of both precision and recall.

Nimishakavi et al. [39] have explored the idea of using tensor factorization models for inducing new relations and their schemas from OpenIE triples into an ontology. By a relation schema, they mean the type signature of the relation. For example, the type signature for the relation $cityLocatedInCountry$ is cityLocatedInCountry(City, Country). The OpenIE triples are represented as a tensor. An element x_{ijk} of the tensor refers to triple formed by i^{th} noun phrase, j^{th} noun phrase and k^{th} verb phrase. The possible hypernyms of the noun phrases are collected from the text corpus using Hearst patterns [23] (for example, "a <hypernym> such as a <noun phrase>") and stored in a matrix. Another matrix is used to store the similarity between the verb phrases (relations). The intuition behind using this similarity matrix is that if two relations are found to be similar in meaning, then their type signatures should also be same/similar. By similarity, the authors mean the cosine similarity of the Word2Vec vectors of the verbs [36]. Coupled factorization of the tensor and the two input matrices is performed to obtain a core tensor which contains the relation schemas such as $suffer\_from(patient, disease)$, $have\_undergo(patient, treatment)$ etc. and a matrix containing the assignment of the noun phrases to the classes.

SOFIE, the system proposed in [46], has been primarily designed for adding more instances of existing relations i.e. for A-Box enrichment. However, the authors have conducted experiments and demonstrated its application for adding a new property and its instances. The authors introduce seed instances manually for a new property and thus adopt the same system to add more instances of the new property. SOFIE works in the following manner: First, facts are collected in two ways - ontological facts collected from the dataset under consideration (includes the manual seed instances for the relation) and textual facts collected from the corpus. These existing facts are given a truth value of 1. Hypotheses for new facts are formed using the known facts. Truth value of these hypotheses are said to be unknown. In order to determine which hypotheses should be accepted as true facts, a set of manually written logical rules are employed. Now the problem is recast as finding the hypotheses that are likely to be true, such that maximal number of rules are satisfied. This can be seen as a maximum satisfiability problem (MAX SAT problem) with all facts, hypotheses and rules rewritten as logical clauses in a uniform manner. A lower set of weights are assigned to clauses which can be violated and a very large weight is assigned for those clauses which are derived from existing facts. A new approximation algorithm (as MAX SAT problem is NP-Hard) called the Functional Max Sat (FMS) algorithm has been implemented to solve this Weighted Max Sat problem. It should be noted that SOFIE is different from the other systems described in this Section in the following aspect: SOFIE needs to know what property should be added to the linked dataset, while the other systems do not take this input.

5 Discovering New Classes

There are quite a few works in the literature which focus on learning class expressions to enrich ontologies. Petrucci et al. [42] solve the problem of class expression learning from natural language text with a learn-by-examples approach. They formulate the problem as a machine transduction task. In this case, a sequence of words in natural language has to be converted into a sequence of logical symbols - a formula. The system operates in two parallel phases, namely, sentence transduction and sentence tagging. The sentence transduction phase identifies the logical structure of the formula corresponding to the natural language input given. The output of this phase is a formula template. The sentence tagging phase tags each word of the input sentence into one of the following types: a concept, a role, a number, or a generic word. Then these tagged words are fit into the formula template to generate the final class expression. For example, let (2) be given as input to both the phases.

$$\textit{A bee is an insect that has 6 legs and produces honey.} \tag{2}$$

Sentence transduction phase outputs the template (3) while the sentence tagging phase tags the sentence and outputs (4).

$$C_0 \sqsubseteq C_1 \sqcap (= N_0 R_0.C_2) \sqcap (\exists R_1.C_3) \tag{3}$$

$$\text{A } [bee]_{C_0} \text{ is an } [insect]_{C_1} \text{ that } [has]_{R_0}[6]_{N_0}[legs]_{C_2} \text{ and } [produces]_{R_1}[honey]_{C_3} \tag{4}$$

The outputs of both the phases are combined to produce the class expression given in (5).

$$Bee \sqsubseteq Insect \sqcap (= 6 have.Leg) \sqcap (\exists produce.Honey) \tag{5}$$

Both the phases employ Recurrent Neural Networks (RNNs) to accomplish their goals. The training data for sentence transduction phase would ideally consist of huge number of pairs of natural language sentences and their corresponding

DL axioms. Since such a dataset was not available, the authors have created such a training dataset. The authors have first verbalized a set of OWL class definitions using Attempto Controlled English (ACE) [26] to get definitions such as the one given in Eq. (2). Then natural language variations of the verbalization were added manually and finally a generalized grammar was built to generate huge number of such training instances.

The authors of [2] handle the problem of class expression learning through syntactic transformation of English sentences to OWL axioms. Syntactic transformation is implemented through various rules of transformation of the parse tree of a sentence. The paper proposes a new controlled natural language called TEDEI (TExtual DEscription Identifier) to define the scope of the input sentences that can be handled by their system. They employ an existing controlled natural language, namely ACE, as an intermediate language and in this way, address some of the limitations of ACE in the context of ontology authoring. They also investigate the impact of two types of ambiguity in natural language sentences, namely lexical ambiguity and semantic ambiguity. Instead of producing one axiom from a given sentence, their system generates all possible axioms that can be generated from the sentence, which are then presented to the user.

As mentioned in Sect. 3, the DL-Learner framework [12] encompasses a set of algorithms for learning class expressions by means of refinement operators i.e. a refinement operator is used to traverse an ordered search space in order to determine the correct concept definition. Informally, a refinement operator can be defined as follows: a downward refinement operator is one which gives rise to a set of more specific concepts and an upward refinement operator returns a set of more general concepts for the given input concept. The general goal of these algorithms is to devise refinement operators that have the following properties [31] while still being able to efficiently traverse through the search space in search of good candidate class expressions:

Let ρ be a downward refinement operator.

Finite: ρ is finite iff $\rho(C)$ is finite for any concept C.
Non-redundant: ρ is redundant iff there exists a refinement chain from a concept C to a concept D, which does not go through some concept E and a refinement chain from C to a concept approximately equal to D, which does go through E.
Proper: ρ is proper iff for all concepts C and D, $D \in \rho(C)$ implies $C \not\equiv D$.
Complete: ρ is complete iff for all concepts C and D with $C \sqsubset D$ we can reach a concept E with $E \equiv C$ from D by ρ
Ideal: ρ is ideal iff ρ is finite, complete, and proper.

However, no refinement operator is ideal and hence the algorithms in the framework work towards handling the missing properties. The major refinement-operator based algorithms are OCEL, CELOE, ELTL and ISLE [12]. OCEL (OWL Class Expression Learner) was the first algorithm defined specifically for the Description Logic ALC. It was designed to cope with redundancy and lack of finiteness property of the refinement operator. CELOE (Class Expression Learning for Ontology Engineering) [29] which is an evolved form of OCEL contains changes specific for learning shorter class expressions as long concept expressions are difficult to maintain and understand in the context of ontology creation. ELTL (EL Tree Learner) [30] is an algorithm for class expression learning specifically designed to suit the OWL EL profile. ISLE (Inductive Statistical Learning of Expressions) [10] which is an extension of the ELTL, also took textual evidence from external corpus into account. Information from the corpus has been used to modify the search heuristic and has been proven to give more accurate expressions on manual evaluation.

Another set of algorithms proposed within the DL-Learner framework for class expression learning which are not based on refinement operators are PARCEL and Fuzzy-DLL. PARCEL (Parallel Class Expression Learning) [50] is suitable for situations which are better solved by parallelization. PARCEL computes partial definitions of a learning problem, which are then aggregated to give complete solutions. Fuzzy-DLL [24] was proposed to handle class expression learning in vague and imprecise domains.

While the above described systems learn class expressions, the system proposed in [39] (see Sect. 4) finds and adds new classes (atomic class names) to the ontology in the process of finding new properties. The coupled tensor factorization process results in a core tensor and a matrix. The core tensor consists of the relation schemas generated and the matrix contains noun phrases assigned to new classes.

5.1 Discussion

The task of inducing new properties and classes from within the linked dataset itself is very difficult to accomplish and hence it becomes imperative to make use of external sources. In this context, data generated through web-scale information extraction systems [17] (which include OpenIE systems such as TextRunner [4], WOE [53], ReVerb [17], SRLIE [14], OLLIE [35] and systems such as NELL, ClausIE [15]) serve as a good starting point for enriching Linked Data. Mapping triples from the former kind of systems (let us call them web triples) to Linked Data's RDF triples can be beneficial in two ways: Linked Data can give more structure and accuracy to the web triples and Linked Data can be enriched (both A-box as well as T-Box) through the web triples. We have seen this trend in [7,38] and also in [39] where the web triples form one of the main inputs for the proposed system. Another set of works following this direction are [16,56]. [56] proposes a framework to give RDF representation to NELL triples. [16] gives RDF representation to NELL KB by linking it to DBpedia and also enriches DBpedia in the process. However, these works are confined mostly to the NELL KB while the opportunities of exploiting the outcomes of the other web-scale IE projects remain largely unexplored.

6 Conclusion

In order to realize the full potential of Linked Data in various applications, it is important to enrich LOD with as many appropriate ontological axioms and assertions as possible. This paper acquaints the readers with the recent advancements in the field of T-Box enrichment of LOD datasets. Techniques for discovery of property and class axioms are mostly based on the RDF triples from within the linked datasets itself while discovery of new properties and classes rely on external sources of data such as OpenIE triples. These enrichment techniques move the datasets towards completeness, all the while making sure that the datasets remain consistent and the manual effort for verifying the correctness of the newly added properties, classes and axioms is reduced. However, as discussed in Sects. 3.1 and 5.1, there are many directions to be explored that might enable further enrichment of LOD.

References

1. Linked Data - Connect Distributed Data across the Web. http://linkeddata.org/
2. Alex Mathews, K., Sreenivasa Kumar, P.: Extracting ontological knowledge from textual descriptions through grammar-based transformation. In: Proceedings of the Ninth International Conference on Knowledge Capture (K-CAP), 4–6 December, Austin, Texas, USA (2017)
3. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, New York (2003)
4. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the web. In: IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, 6–12 January 2007, pp. 2670–2676 (2007)
5. Barati, M., Bai, Q., Liu, Q.: Mining semantic association rules from RDF data. Knowl. Based Syst. **133**, 183–196 (2017)
6. Barchi, P.H., Hruschka, E.R.: Never-ending ontology extension through machine reading. In: 2014 14th International Conference on Hybrid Intelligent Systems, pp. 266–272, December 2014
7. Barchi, P.H., Hruschka, E.R.: Two different approaches to ontology extension through machine reading. J. Netw. Innov. Comput. **3**(1), 78–87 (2015)
8. Basse, A., Gandon, F., Mirbel, I., Lo, M.: DFS-based frequent graph pattern extraction to characterize the content of RDF triple stores. In: Web Science Conference 2010 (WebSci 2010) (2010)
9. Borgelt, C., Kruse, R.: Induction of association rules: apriori implementation. In: Härdle, W., Rönz, B. (eds.) Compstat, pp. 395–400. Springer, Heidelberg (2002). https://doi.org/10.1007/978-3-642-57489-4_59
10. Bühmann, L., Fleischhacker, D., Lehmann, J., Melo, A., Völker, J.: Inductive lexical learning of class expressions. In: Janowicz, K., Schlobach, S., Lambrix, P., Hyvönen, E. (eds.) EKAW 2014. LNCS (LNAI), vol. 8876, pp. 42–53. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13704-9_4
11. Bühmann, L., Lehmann, J.: Pattern based knowledge base enrichment. In: Alani, H., et al. (eds.) ISWC 2013. LNCS, vol. 8218, pp. 33–48. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41335-3_3

12. Bühmann, L., Lehmann, J., Westphal, P.: DL-Learner - a framework for inductive learning on the semantic web. Web Semant. Sci. Serv. Agents WWW **39**, 15–24 (2016)
13. Cergani, E., Miettinen, P.: Discovering relations using matrix factorization methods. In: 22nd ACM International Conference on Information and Knowledge Management, CIKM 2013, San Francisco, CA, USA, 27 October-1 November, 2013, pp. 1549–1552 (2013)
14. Christensen, J., Mausam, Soderland, S., Etzioni, O.: An analysis of open information extraction based on semantic role labeling. In: Proceedings of the 6th International Conference on Knowledge Capture (K-CAP 2011), 26–29 June, 2011, Banff, Alberta, Canada, pp. 113–120 (2011)
15. Del Corro, L., Gemulla, R.: ClausIE: clause-based open information extraction. In: Proceedings of the 22nd International Conference on World Wide Web, WWW 2013, pp. 355–366 (2013)
16. Dutta, A., Meilicke, C., Stuckenschmidt, H.: Semantifying triples from open information extraction systems. In: STAIRS 2014 - Proceedings of the 7th European Starting AI Researcher Symposium, Prague, Czech Republic, 18–22 August 2014, pp. 111–120 (2014)
17. Etzioni, O., Fader, A., Christensen, J., Soderland, S., Mausam, M.: Open information extraction: the second generation. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - IJCAI 2011, vol. 1, pp. 3–10. AAAI Press (2011)
18. Fleischhacker, D., Völker, J.: Inductive learning of disjointness axioms. In: Meersman, R., et al. (eds.) OTM 2011. LNCS, vol. 7045, pp. 680–697. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25106-1_20
19. Fleischhacker, D., Völker, J., Stuckenschmidt, H.: Mining RDF data for property axioms. In: Meersman, R., et al. (eds.) OTM 2012. LNCS, vol. 7566, pp. 718–735. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33615-7_18
20. Galárraga, L., Heitz, G., Murphy, K., Suchanek, F.M.: Canonicalizing open knowledge bases. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 1679–1688. ACM (2014)
21. Galárraga, L.A., Preda, N., Suchanek, F.M.: Mining rules to align knowledge bases. In: Proceedings of the 2013 Workshop on Automated Knowledge Base Construction, pp. 43–48. ACM (2013)
22. Galárraga, L.A., Teflioudi, C., Hose, K., Suchanek, F.: AMIE: Association rule Mining under Incomplete Evidence in ontological knowledge bases. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 413–422. ACM (2013)
23. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: 14th International Conference on Computational Linguistics, COLING 1992, Nantes, France, 23–28 August 1992, pp. 539–545 (1992)
24. Iglesias, J., Lehmann, J.: Towards integrating fuzzy logic capabilities into an ontology-based inductive logic programming framework. In: 2011 11th International Conference on Intelligent Systems Design and Applications, pp. 1323–1328, November 2011
25. Irny, R., Kumar, S.P.: Mining inverse and symmetric axioms in Linked Data. In: Proceedings of the Seventh Joint International Semantic Technologies Conference, Gold Coast, Australia, 10–12 November (2017)
26. Kaljurand, K., Fuchs, N.E.: Verbalizing OWL in Attempto Controlled English. In: Proceedings of the OWLED 2007 Workshop on OWL: Experiences and Directions, Innsbruck, Austria, 6–7 June 2007 (2007)

27. Kasneci, G., Elbassuoni, S., Weikum, G.: MING: mining informative entity relationship subgraphs. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 1653–1656. ACM (2009)
28. Koutraki, M., Preda, N., Vodislav, D.: Online relation alignment for linked datasets. In: Blomqvist, E., Maynard, D., Gangemi, A., Hoekstra, R., Hitzler, P., Hartig, O. (eds.) ESWC 2017. LNCS, vol. 10249, pp. 152–168. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58068-5_10
29. Lehmann, J., Auer, S., Bühmann, L., Tramp, S.: Class expression learning for ontology engineering. J. Web Semant. **9**(1), 71–81 (2011)
30. Lehmann, J., Haase, C.: Ideal downward refinement in the $\mathcal{EL}$ description logic. In: De Raedt, L. (ed.) ILP 2009. LNCS (LNAI), vol. 5989, pp. 73–87. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13840-9_8
31. Lehmann, J., Hitzler, P.: Foundations of refinement operators for description logics. In: Blockeel, H., Ramon, J., Shavlik, J., Tadepalli, P. (eds.) ILP 2007. LNCS (LNAI), vol. 4894, pp. 161–174. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78469-2_18
32. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: DBpedia - a large-scale, multilingual knowledge base extracted from Wikipedia. Semant. Web **6**, 167–195 (2015)
33. Li, H., Sima, Q.: Parallel mining of OWL 2 EL ontology from large linked datasets. Knowl. Based Syst. **84**, 10–17 (2015)
34. Mahdisoltani, F., Biega, J., Suchanek, F.M.: YAGO3: a knowledge base from multilingual Wikipedias. In: CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, 4–7 January 2015, Online Proceedings (2015)
35. Mausam, M.S., Bart, R., Soderland, S., Etzioni, O.: Open language learning for information extraction. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, pp. 523–534 (2012)
36. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 26, pp. 3111–3119 (2013)
37. Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B., Krishnamurthy, J., Lao, N., Mazaitis, K., Mohamed, T., Nakashole, N., Platanios, E., Ritter, A., Samadi, M., Settles, B., Wang, R., Wijaya, D., Gupta, A., Chen, X., Saparov, A., Greaves, M., Welling, J.: Never-ending learning. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2015) (2015)
38. Mohamed, T.P., Hruschka Jr., E.R., Mitchell, T.M.: Discovering relations between noun categories. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, pp. 1447–1455 (2011)
39. Nimishakavi, M., Saini, U.S., Talukdar, P.P.: Relation schema induction using tensor factorization with side information. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, 1–4 November 2016, pp. 414–423 (2016)
40. Papadakis, G., Ioannou, E., Niederée, C., Fankhauser, P.: Efficient entity resolution for large heterogeneous information spaces. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 535–544. ACM (2011)

41. Paulheim, H.: Knowledge graph refinement: a survey of approaches and evaluation methods. Semant. Web **8**(3), 489–508 (2017)
42. Petrucci, G., Ghidini, C., Rospocher, M.: Ontology learning in the deep. In: Blomqvist, E., Ciancarini, P., Poggi, F., Vitali, F. (eds.) EKAW 2016. LNCS (LNAI), vol. 10024, pp. 480–495. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49004-5_31
43. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. IEEE Trans. Knowl. Data Eng. **25**(1), 158–176 (2013)
44. Subhashree, S., Kumar, P.S.: Enriching linked datasets with new object properties. CoRR abs/1606.07572 (2016). http://arxiv.org/abs/1606.07572
45. Suchanek, F.M., Abiteboul, S., Senellart, P.: PARIS: probabilistic alignment of relations, instances, and schema. Proc. VLDB Endow. **5**(3), 157–168 (2011)
46. Suchanek, F.M., Sozio, M., Weikum, G.: SOFIE: a self-organizing framework for information extraction. In: Proceedings of the 18th International Conference on World Wide Web, WWW 2009, New York, pp. 631–640. ACM (2009)
47. Thor, A., Anderson, P., Raschid, L., Navlakha, S., Saha, B., Khuller, S., Zhang, X.-N.: Link prediction for annotation graphs using graph summarization. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011. LNCS, vol. 7031, pp. 714–729. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25073-6_45
48. Tonon, A., Catasta, M., Demartini, G., Cudré-Mauroux, P.: Fixing the domain and range of properties in Linked Data by context disambiguation. In: LDOW@ WWW (2015)
49. Töpper, G., Knuth, M., Sack, H.: DBpedia ontology enrichment for inconsistency detection. In: Proceedings of the 8th International Conference on Semantic Systems, pp. 33–40. ACM (2012)
50. Tran, A.C., Dietrich, J., Guesgen, H.W., Marsland, S.: An approach to parallel class expression learning. In: Bikakis, A., Giurca, A. (eds.) RuleML 2012. LNCS, vol. 7438, pp. 302–316. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32689-9_25
51. Völker, J., Niepert, M.: Statistical schema induction. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011. LNCS, vol. 6643, pp. 124–138. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21034-1_9
52. Wijaya, D., Talukdar, P.P., Mitchell, T.: PIDGIN: ontology alignment using web text as interlingua. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, pp. 589–598. ACM (2013)
53. Wu, F., Weld, D.S.: Open information extraction using Wikipedia. In: ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 11–16 July 2010, Uppsala, Sweden, pp. 118–127 (2010)
54. Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., Auer, S.: Quality assessment for Linked Data: a survey. Semant. Web **7**(1), 63–93 (2016)
55. Zheng, W., Zou, L., Peng, W., Yan, X., Song, S., Zhao, D.: Semantic SPARQL similarity search over RDF knowledge graphs. Proc. VLDB Endow. **9**(11), 840–851 (2016)
56. Zimmermann, A., Gravier, C., Subercaze, J., Cruzille, Q.: Nell2rdf: read the web, and turn it into RDF. In: Proceedings of the Second International Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data, Montpellier, France, 26 May 2013, pp. 2–8 (2013)