Moving beyond sameAs with PLATO: Partonomy detection for Linked Data

Prateek Jain
Kno.e.sis Center
Wright State University
Dayton, OH, USA
prateek@knoesis.org

Pascal Hitzler
Kno.e.sis Center
Wright State University
Dayton, OH, USA
pascal.hitzler@wright.edu

Kunal Verma
Accenture Technology Labs
50 West San Fernando Street
San Jose, CA, USA
k.verma@accenture.com

Peter Z. Yeh
Accenture Technology Labs
50 West San Fernando Street
San Jose, CA, USA
peter.z.yeh@accenture.com

Amit Sheth
Kno.e.sis Center
Wright State University
Dayton, OH, USA
amit@knoesis.org

ABSTRACT

The Linked Open Data (LOD) Cloud has gained significant traction over the past few years. With over 275 interlinked datasets across diverse domains such as life science, geography, politics, and more, the LOD Cloud has the potential to support a variety of applications ranging from open domain question answering to drug discovery.

Despite its significant size (approx. 30 billion triples), the data is relatively sparely interlinked (approx. 400 million links). A semantically richer LOD Cloud is needed to fully realize its potential. Data in the LOD Cloud are currently interlinked mainly via the owl:sameAs property, which is inadequate for many applications. Additional properties capturing relations based on causality or partonomy are needed to enable the answering of complex questions and to support applications.

In this paper, we present a solution to enrich the LOD Cloud by automatically detecting partonomic relationships, which are well-established, fundamental properties grounded in linguistics and philosophy. We empirically evaluate our solution across several domains, and show that our approach performs well on detecting partonomic properties between LOD Cloud data.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Learning; H.3.3 [**Information Storage and Retrieval**]: Online Information Services—*web-based services*

Keywords

Part of Relation, Mereology, Linked Open Data Cloud

HT'12, June 25–28, 2012, Milwaukee, Wisconsin, USA.
Copyright 2012 ACM 978-1-4503-1335-3/12/06 ...$10.00.

1. INTRODUCTION

The LOD Cloud consists of datasets linked primarily by the owl:sameAs property created by different organizations. This has proven to be useful for a number of use cases [4, 15], which combine data from multiple ontologies. The current mechanism for linking entities across datasets is using the *sameAs* relationship to assert that two entities are the same. We believe that using the *sameAs* relationship is not sufficient to capture the rich set of relationships between entities. There are a number of other relationships such as partonomy (part-of), and causality [28], whose presence could allow creating even more intelligent applications such as more sophisticated question answering systems like Watson [12]. One of the main reasons why these relationships are not captured is the issue of scale. As there are millions of entities involved, it is a non-trivial task to manually assert these relationships. While there is some level of automation available for creating the *sameAs* links, there is no automation for creating other kinds of relationships [19].

In this paper, we present *PLATO* (**P**art-Of relation finder on **L**inked Open D**A**ta **TO**ol)[1] for automatically creating part-of relationship between entities in the LOD cloud.

We chose part-of relationship for two reasons: 1) it is a well studied field. In particular we use the partonomy classification created by Winston [33] to guide our work and 2) part-of relationships are freely available on the Web in sources such as Wikipedia. The fundamental premise behind our approach is that the web can be mined to automatically detect part-of relationships between entities. Our approach consists of a combination of heuristics for detecting candidate relationships between any two entities. These heuristics range from detecting bi-directionality of links between articles about these entities to ensuring that the involved entities satisfy domain and range constraints of the relevant partonomic relation. The Web is then mined for evidence to support the candidate relationships with the help of pattern based querying. Using this approach, PLATO is able to discover partonomic relationships between entities in the LOD cloud. For example, PLATO was correctly able to discover that Kurt Cobain was a member of the band Nirvana and that Baked Alaska has ice cream as an ingredient.

[1] http://wiki.knoesis.org/index.php/PLATO

These relationships can prove to be extremely useful for the LOD cloud. For example, consider the following query from the National Geographic Bee, ”In which county can you find the village of Crook that is full of lakes?”. The answer for this query can be successfully retrieved using information present in the LOD cloud dataset (e.g. Geonames), if part-of relationships have been identified and asserted within and between datasets [20].

The key contributions of our work are: 1) To the best of our knowledge, PLATO is the first effort on the automatic detection of part-of relationships in the context of the LOD cloud. 2) We believe that PLATO's approach of mining the Web to detect and validate the relationships for LOD cloud is rather unique and thus extends the existing arsenal of ontology engineering methods. 3) We provide a formal representation of the partonomy classification created by Winston. We furthermore present a comprehensive evaluation in which we automatically detect part-of relationships between hundreds of entities from prominent ontologies in the LOD cloud such as DBpedia and Freebase. We also present precision and recall for our partonomy extraction approach, and the results make us believe ours is a practically useful approach.

The rest of the paper is organized as follows: In Section 2 we present Winston's approach to part-of relation and its conversion into an OWL 2 ontology. In Section 3, we present the PLATO approach, followed by a comprehensive evaluation. We then present the related work, future work and conclusion.

2. WINSTON'S APPROACH TO PART-OF RELATIONSHIPS—ONTOLOGIZED

All entities are fundamentally part of some other entity. Researchers in a number of areas, including philosophy [33, 2], linguistics [14] and geographical information systems (GIS) [29, 20, 7] have investigated partonomy. Our work of identification of partonomic relationships between entities uses well-accepted partonomic relationships, which identify the relationships based on the 'type' of entities involved. The part-whole relation, or partonomy, is an important fundamental relationship which manifests itself across all physical entities such as human made objects (Cup-Handle), social groups (Jurors-Jury) and conceptual entities such as time intervals (5th hour of the day). Its frequent occurrence results in a manifestation of a part-for-whole mismatch and whole-for-part mismatch within many domains, and especially in spatial datasets.

Winston [33] created a categorization of part-whole relations which identifies and covers part-whole relations from a number of domains such as artifacts, geographical entities, food and liquids. It is recognized as one of the most comprehensive categorizations of partonomic relationships, and other work in similar spirit such as [13] analyze his categorization.

Winston's categorization has been created using three relational elements:

1. Functional/Non-Functional (F/NF): Parts are in a specific spatial/temporal relationship with respect to each other and to the whole to which they belong. Example: Belgium is a part of NATO partly because of its specific spatial position.
2. Homeomerous/Non-Homeomerous (H/NH): Parts are the same as each other and as the whole. Example: A slice of a pie is the same as other slices and as the pie itself.
3. Separable/Inseparable (S/IN): Parts are separable/ inseparable from the whole. Example: A card can be separated from the deck to which it belongs.

Table 1 illustrates six different types of partonomic relationships based on this categorization, taken from [33], their description using the relational elements and examples of partonomic relationships covered by them.

Using this classification and relational elements, relations between two entities can be marked as partonomic or non-partonomic in nature. If they are partonomic, the category to which they belong can be identified.

In order to use Winston's approach in a Semantic Web context, which is essentially linguistic in nature, we must formalize it by carrying it over to a Semantic Web ontology language. We will thus cast his categorization into an OWL 2 ontology [17] which can then be used in conjunction with a knowledge base of partonomic (and other) information. Let us remark that in [27] a set of best practices have been laid down to deal with straightforward cases for defining classes involving part-whole relations. However their modeling approach is considerably less fine-grained than the one in [33] which we follow here.

For this purpose, we introduce the following OWL property names, which correspond to those listed in Table 1.

- component-integral object: po-component
- member-collection: po-member
- portion-mass: po-portion
- stuff-object: po-stuff
- feature-activity: po-feature
- place-area: po-place

We also use spatially-located-in as the spatial (topological) located-in relationship mentioned in [33], and part-of as the generic part-of (part-whole) relation.

The following axioms can then be drawn from [33]. Let $\mathsf{PO} = \{\mathsf{po\text{-}component}, \mathsf{po\text{-}member}, \mathsf{po\text{-}portion}, \mathsf{po\text{-}stuff}, \mathsf{po\text{-}feature}, \mathsf{po\text{-}place}\}$.

(P1) [33, Section 5] For all $R \in \mathsf{PO}$, R is transitive, asymmetric, and irreflexive (i.e., a strict partial order).

(P2) For all $R \in \mathsf{PO}$, $R \sqsubseteq \mathsf{part\text{-}of}$. Note that this does *not* imply that part-of is transitive, as prescribed in [33].

(P3) spatially-located-in is transitive and reflexive. Note that spatially-located-in should not be understood to be a subproperty of part-of according to [33].

(P4) [33, Section 6] For all $R \in \mathsf{PO}$, we have

$$R \circ \mathsf{spatially\text{-}located\text{-}in} \sqsubseteq \mathsf{spatially\text{-}located\text{-}in} \quad \text{and}$$
$$\mathsf{spatially\text{-}located\text{-}in} \circ R \sqsubseteq \mathsf{spatially\text{-}located\text{-}in}.$$

(P5) [33, page 435] For all $R \in \mathsf{PO} \cup \{\mathsf{spatially\text{-}located\text{-}in}\}$, and all classes C, we have the first-order predicate logic axiom

$$(\forall x)(\forall y)(R(x, y) \wedge C(y) \to (\exists z)(R(x, z) \wedge C(z)).$$

Note that this is a tautology.

Category	Description	Example	Text Patterns
Component-Integral Object	Parts are functional, non-homeomerous and separable from the whole.	Handle-Cup	part of, component of
Member-Collection	Parts are non functional, non homeomerous and separable from the whole.	Tree-Forest	member of, part-of
Portion-Mass	Parts are non-functional, homeomerous and separable from the whole.	Slice-Pie	of, part-of
Stuff-Object	Parts are non-functional, non-homeomerous and inseparable from the whole.	Gin-Martini	is partly, made of
Feature-Activity	Parts are functional, non-homeomerous and inseparable from the whole.	Paying-Shopping	has, have
Place-Area	Parts are non-functional, homeomerous and inseparable from the whole.	Everglades-Florida	located in, part-of

Table 1: Six type of partonomic relation with relational elements

(P6) [33, page 435] For all $R \in \textsf{PO} \cup \{\textsf{spatially-located-in}\}$, and all classes C, we have the first-order predicate logic axiom

$$(\forall x)(\forall y)(C(y) \land (C(y) \to R(x,y)) \to R(x,y)).$$

Please note that this is a tautology.

Summarizing, we can axiomatize (P1) to (P4) as the following axioms—we will discuss (P5) and (P6) further below.

- For all $R \in \textsf{PO}$, R is transitive, antisymmetric, and irreflexive.
- For all $R \in \textsf{PO}$, $R \sqsubseteq \textsf{part-of}$.
- $\textsf{spatially-located-in}$ is transitive and reflexive.
- For all $R \in \textsf{PO}$, we have

  $$R \circ \textsf{spatially-located-in} \sqsubseteq \textsf{spatially-located-in} \quad \text{and}$$
  $$\textsf{spatially-located-in} \circ R \sqsubseteq \textsf{spatially-located-in}.$$

This results in a total of $3 \cdot 6 + 2 \cdot 6 + 2 + 6 \cdot 2 = 44$ axioms, all expressible in OWL 2.

However, there is a catch. While all these axioms are expressible in OWL 2 (more precisely, in OWL 2 Full), the collection of these ontologies does not constitute a valid OWL 2 DL ontology. The reason for this is that (P1) violates a global constraint on OWL 2 DL ontologies given in [24, Section 11]: A property cannot be transitive and irreflexive at the same time.[2] In other words, we cannot specify strict partial orders in OWL 2 DL.[3] The most straightforward way to fix this, is to drop one of the requirements on R in (P1), and the most obvious candidate would be to drop the irreflexivity axioms. The resulting set of 38 axioms then constitutes a valid OWL 2 DL ontology.

Let us now return to the axioms from (P5) and (P6). They are tautologies in first-order predicate logic, which means that they do not contribute any additional knowledge. As such, they do not need to be added to our ontology.[4] Note that this does not mean that the observations leading to (P5) and (P6) in [33] are void: We obtain tautologies because the use of OWL suggests a particular type of modeling class membership (called class inclusion in [33]) which is probably not obvious or necessary from a more general, linguistic perspective.

It is possible to partially recover irreflexivity of the $R \in \textsf{PO}$. One way to do this is to use the DL-safe SWRL rule [18, 21, 25] $R(x,y) \land R(y,x) \to x \neq y$, which expresses the same as irreflexivity, however its application is restricted to known individuals and is thus weaker than (first-order logic) irreflexivity. Another alternative is to use nominal schemas [21, 22], e.g. by means of the axiom[5]

$$\{x\} \sqcap \exists R.\exists R.\{x\} \sqsubseteq \bot$$

which can actually be understood as a macro that results in n OWL 2 DL axioms, where n is the number of known individuals in the knowledge base.[6]. This means that we can incorporate a weak form of irreflexivity *in OWL 2 DL* without having to use DL-safe SWRL (and software which supports the latter).

There is yet another catch: All properties occurring in the above constructed part-of ontology are complex (i.e., non-simple), and OWL 2 DL has global restrictions on the use

[2] A transitive property is complex, and thus not simple. However only simple properties are allowed to be irreflexive.

[3] Note that transitivity and irreflexivity of a property R imply that R is also antisymmetric (i.e., a strict partial order): Assume R were transitive and irreflexive, but not antisymmetric. Then, because R is not antisymmetric we must have a, b with $R(a,b)$ and $R(b,a)$ and $a \neq b$. But by transitivity of R, we obtain $R(a,a)$ from $R(a,b)$ and $R(b,a)$ which is impossible by irreflexivity.

[4] In other words, adding them would accomplish nothing.

[5] Nominal schemas could also be used to directly express the just mentioned DL-safe rule [22]. However, this would result in a more complicated axiom with two nominal schemas, which is less favorable in terms of scalability.

[6] The OWL 2 DL axioms are obtained by *grounding*: Replace $\{x\}$ by all available nominals $\{a\}$, a being a known individual, each such replacement resulting in one OWL 2 DL axiom.

of such properties. If this ontology is used in conjunction with a domain ontology, then these global restrictions may be violated. Likewise, usage of properties in OWL 2 DL is globally restricted by the so-called *regularity* condition,[7] which may also be violated if the part-of ontology is used together with a domain ontology. In a way similar to the irreflexivity issue discussed above, it is possible to recover from this by expressing some (or all) of the axioms in the part-of ontology in weaker form, using DL-safe rules or nominal schemas. How this is best done depends on the domain ontology, but it is always possible in principle, and indeed relatively straightforward.

3. APPROACH

Given a LOD Cloud dataset, our solution – PLATO – automatically enriches it with partonomy properties through four key steps.[8]

First, PLATO generates candidate pairs of entities from the dataset. Second, PLATO generates "hypothesis" of possible partonomy properties – represented as linguistic patterns – for each entity pair. Next, PLATO tests the resulting patterns (and hence hypotheses) in a corpus driven manner. Finally, PLATO asserts only those partonomy properties with strong supporting evidence. Figure 1 depicts the workflow, which we describe in more detail in the subsequent sections.

3.1 Candidate Generation

Given a LOD Cloud dataset, PLATO generates all possible pairs between the entities in the dataset. However, the number of entity pairs can be extremely large, which can make the subsequent steps intractable. To address this problem, PLATO filters unpromising entity pairs using a simple heuristic—i.e. entities that are strongly associated are more likely to be related via some property than those that are not. PLATO implements this heuristic by exploiting Wikipedia. The references between Wikipedia pages provide a good proxy for association. Moreover, Wikipedia provides comprehensive coverage across diverse domains. For each entity pair, PLATO retrieves the corresponding Wikipedia page of each entity—using the Mediawiki API[9]—and if these pages refer to each other, then the pair is said to be strongly associated and kept for subsequent processing. Otherwise, the pair is discarded.

For datasets besides DBpedia, such as Freebase, we use the sameAs links present between DBpedia entity (e.g. dbpedia: Cellulose) and entity of other datasets (e.g. fbase: Cellulose). Then we check if the any of the entity refers to the other one. For example, if fbase: Chicken links to dbpedia: Salt. This is just a way to reduce the number of candidate pairs and it is possible to use other techniques to generate these pairs. The use of dataset specific heuristics has been used in other tools such as SILK [31], in order to maximize finding relationships between any two datasets. It is possible to replace this module with another heuristics to generate candidate pairs and use the rest of the system without any modifications.

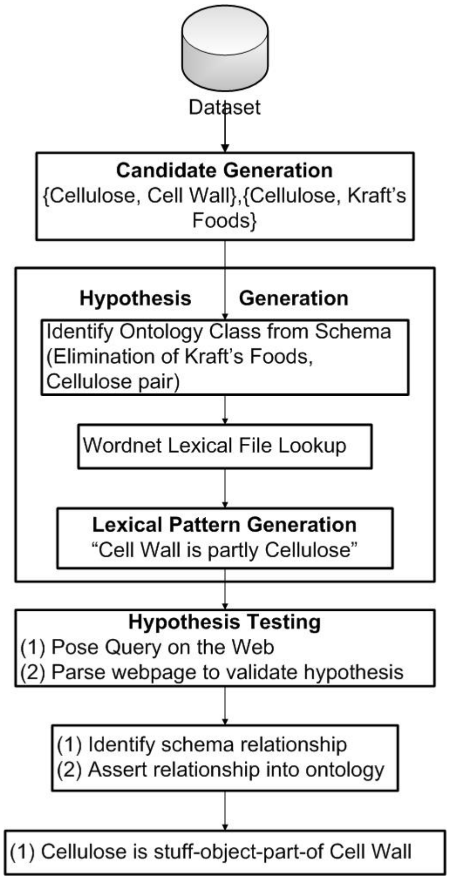
Dataset
Candidate Generation
{Cellulose, Cell Wall},{Cellulose, Kraft's Foods}
Hypothesis Generation
Identify Ontology Class from Schema (Elimination of Kraft's Foods, Cellulose pair)
Wordnet Lexical File Lookup
Lexical Pattern Generation
"Cell Wall is partly Cellulose"
Hypothesis Testing
(1) Pose Query on the Web
(2) Parse webpage to validate hypothesis
(1) Identify schema relationship
(2) Assert relationship into ontology
(1) Cellulose is stuff-object-part-of Cell Wall

Figure 1: PLATO system flow chart

Please note, in principal it is possible to replace the usage of Mediawiki API with entities directly from DBpedia. However, it may result in the loss of some useful candidate pairs as DBpedia captures limited information from Wikipedia. For example, as of 6th February 2012, the DBpedia page for Cellulose does not refer to Carbon. However, the Wikipedia pages for Carbon and Cellulose do refer to each other, thus making them possible candidate pairs for consideration.

For example, given the DBpedia dataset from the LOD Cloud, some of the entity pairs generated by PLATO will include:

- Cellulose, Cell Wall
- Cellulose, Kraft's Food

PLATO retrieves the Wikipedia pages for Cellulose, Cell Wall, and Kraft's Foods. The Wikipedia pages for Cellulose and Cell Wall refer to each other, so this pair is kept. The Wikipedia page for Cellulose refers to the page for Kraft's Foods, due to usage of Cellulose in cheese manufacturing at Kraft's Foods. However, the page for The Kraft's Foods does not refer back to the page for Cellulose. Hence, this pair is considered to be only weakly associated by PLATO, and thus discarded.

3.2 Hypothesis Generation

PLATO generates hypotheses of possible OWL partonomy properties (described in Section 2) for each entity pair

[7] See "Restriction on the Property Hierarchy" in [24, Section 11].

[8] PLATO follows these same four steps for enriching multiple LOD Cloud datasets. For ease of exposition, we will describe PLATO in the context of enriching a single dataset.

[9] http://en.wikipedia.org/w/api.php

from the previous step. PLATO now determines the type of each entity in the pair using WordNet [11]—a lexical taxonomy that is well suited for this task. Specifically, PLATO retrieves the lexicographer file of the WordNet synset corresponding to each entity to serve as its type.[10] The name of this file has the form *POS.SUFFIX* where *POS* is the part-of-speech (i.e. noun, verb, adv, or adj) and *SUFFIX* is the broader group that the synset (and hence entity) belongs to (e.g. animal, plant, etc.). For example, given the entity pair (*Cell Wall*, *Cellulose*), lexicographer files of the synsets corresponding to these entities are both noun.body.

PLATO uses this information to determine the applicable OWL partonomy properties. We captured these properties from Winston's taxonomy of part-whole relations [33] (see Section 2), which was chosen for the following reasons:

- Winston's taxonomy is well-established and widely accepted.
- Winston provides guidelines on what types are applicable to each part-whole relationship—e.g. Winston's *Place-Area* relationship applies to only areas, places, and locations. These guidelines can be captured as domain-range axioms for each corresponding OWL partonomy property.
- Winston suggests linguistic cues for each part-whole relationship, which PLATO can use to generate linguistic patterns.

If *POS* is not a noun or verb, then PLATO discards the entity pair because Winton's relationships apply to only nouns and verbs. If so, then PLATO uses the SUFFIX to determine the OWL partonomy properties that are applicable based on their domain and range. Returning to our example, the OWL properties of po-component and po-stuff—corresponding to Winston's *Component-Integral-Object* and *Stuff-Object* relationships respectively—are applicable because the SUFFIXES of Cell Wall and Cellulose satisfy the domain and range of these properties.

Finally, PLATO generates linguistic patterns for each applicable property based on linguistic cues suggested by Winston. For example, the linguistic cues for po-stuff include "is made of" and "is partly." From these cues, the following linguistic patterns are generated for *(Cell Wall, Cellulose)*:

- Cell Wall is made of Cellulose
- Cellulose is made of Cell Wall
- Cell Wall is partly Cellulose
- Cellulose is partly Cell Wall

These patterns serve as hypotheses to be validated in the next step.

3.3 Hypothesis Testing

PLATO tests the lexical patterns for each entity pair in a corpus-driven manner. PLATO uses the Web as the corpus because of its coverage, and uses publicly available search APIs to access its contents. Specifically, PLATO uses the Bing Search API 2.0[11] because it allows unlimited searches.

For each pattern generated for an entity pair, PLATO executes a search of the pattern using the BING API, and takes the top N search results (i.e. URLs for the top N webpages) returned by BING. N can be adjusted by the user; and PLATO sets the default value of N to 50, which we found to produce good results empirically. For each resulting URL, PLATO fetches the page it points to—using off-the-shelf crawling and html parsing technologies, e.g., JSOUP[12]—and determines whether the pattern appears in the page based on exact string match with stemming. This step is necessary because the search results can contain spurious pages—i.e. pages that do not contain the actual pattern. For example, a page containing the string "Is the cell wall of a plant made of cellulose fibers?" may appear in the search result for the pattern "cell wall is made of cellulose"; but this string does not match the pattern (and hence does not support it). The crawling of the page is necessary as the snippet of the page in the result is typically retrieved from the cache, and the actual content may or may not reflect the same content.

Finally, PLATO counts the total number of pages that contain the pattern, and uses this count as the level of support for the OWL partonomy property—associated with the pattern—that could exist between the entity pair. For each entity pair, PLATO asserts the partonomy property whose associated pattern has the strongest supporting evidence, computed from the previous step. Returning to our example for the entity pair (*Cell Wall*, *Cellulose*), the supporting evidence for each pattern associated with the pair (assuming a search limit of 50) is below:

- Cell Wall is made of Cellulose, 48
- Cellulose is made of Cell Wall, 10
- Cell Wall is partly Cellulose, 50
- Cellulose is partly Cell Wall, 7

Since the pattern 'Cell Wall is partly Cellulose' has the strongest support, the associated property po-stuff—corresponding to Winston's *Stuff-Object* relationships—is asserted, with Cellulose as the part and Cell Wall as the whole.

In addition to adding properties at the instance-level (i.e. between entities), PLATO also enriches the schema by generalizing from the instance level assertions. To explain this step, let C and D be two classes about which we want to find out whether they should be related on the schema level by one of the partonomic relationships R. From the process just described, we obtain a set $M_{R,C,D}$ of instance level assertions of the form $R(a,b)$, where $a \in C$ and $b \in D$.[13] We now add schema level axioms according to the following rules: (1) If, for all $a \in C$, there is a $b \in D$ with $R(a,b) \in M_{R,C,D}$, then add the axiom $C \sqsubseteq \exists R.D$, which can be expressed in OWL/RDF serialization using the *owl:someValuesFrom* property restriction. (2) If, for all $b \in D$, there is a $a \in C$ with $R(a,b) \in M_{R,C,D}$, then add the axiom $D \sqsubseteq \exists R^{-}.C$, were R^{-} indicates the inverse (using *owl:inverseOf*) property of R. While this approach seems to be rather crude

[10] If a WordNet synset cannot be found for an entity, then PLATO will generalize the entity by looking up its superclass in DBpedia using the JENA ARQ API (http://openjena.org/).

[11] http://msdn.microsoft.com/en-us/library/dd251056.aspx

[12] http://jsoup.org/apidocs/

[13] If we did not obtain any such assertion, then we do not add any schema axiom.

compared to schema learning methods based on inductive paradigms,[14] it already achieves good results, as can be seen from our evaluation in Section 4.3.

4. EVALUATION

We present three experiments to evaluate the performance of PLATO on enriching LOD Cloud dataset with partonomy properties. The first experiment evaluates PLATO's performance on discovering partonomy properties between entities within the same LOD Cloud dataset (i.e. intra-dataset instance-level partonomy discovery). The second experiment evaluates PLATO's performance across different LOD Cloud datasets (i.e. inter-dataset instance-level partonomy discovery). The final experiment evaluates PLATO's performance on discovery partonomy properties at the schema level. All the evaluation components of this work are available for download at the PLATO Project Page[15]

4.1 Intra-Dataset Instance-Level Partonomy Discovery

We evaluated the performance of PLATO on discovering partonomy properties between entities within the same LOD Cloud dataset using the following methodology. First, we chose the DBpedia dataset because: 1) it is one of the largest datasets available on the Linked Open Data Cloud; and 2) it covers diverse domains such as Geography, Science, Politics, History and Arts [5]. The scale and coverage of DBpedia allows us to thoroughly evaluate the performance of PLATO across different partonomy types [33] and domains.

Next, we randomly generated 83,639 entity pairs from DBpedia for evaluation because it was not practical to generate all possible entity pairs given DBpedia's size. We used the Mediawiki API[16] to randomly generate a pair of Wikipedia articles, whose URLs were then translated to the corresponding DBpedia entities. Given that it is not practical to generate all entity pairs within DBpedia, this method provides an unbiased dataset for evaluation.

We then applied PLATO to the resulting dataset to automatically discover partonomy properties between each entity pair. For each partonomy property discovered, the property was randomly assigned to one of three human graders, who validated its correctness. A human grader determined that the partonomy property discovered by PLATO between a pair of entities is correct if the following conditions are all satisfied:

- A part-whole relationship does exist between the entities
- The correct partonomy property is given
- The part-whole roles are correctly assigned to the entities – e.g., given the pair cell and cell wall, cell is the whole and cell wall is the part.

Finally, we report the precision (i.e. the number of correct partonomy properties discovered by PLATO over the total number of partonomy properties discovered) based on the human grader's responses. We did not report the recall for PLATO because: 1) an existing DBpedia benchmark for this purpose does not exist, and 2) the large number of entity pairs made it difficult to compute the recall manually due to time and resource limitations.

Table 2 shows the results for this experiment. Of the 83,639 entity pairs generated, PLATO discovered partonomy properties for 13,853 pairs. We should note that partonomy relationships do not exist for many of the entity pairs because these pairs were randomly generated – e.g. a random sample of 100 pairs found only 11 to have a valid partonomy relationship. PLATO was able to filter many of these extraneous pairs based on the heuristic that two entities must be strongly associated (see Section 3.1). Overall, PLATO achieved high precision in discovering partonomy properties between entities in DBpedia. Moreover, PLATO discovered partonomy properties across a wide range of entities ranging from places to chemical compounds. However, PLATO did have low precision for a couple of partonomy properties – i.e. 'Portion-Mass' and 'Place-Area'. For 'Portion-Mass', PLATO did not find any entities related to each other. This is understandable as this property deals with very abstract entities such as 'Slice of Lemon', 'Hunk of Clay', etc. and hence it's hard to find entities of this type in DBpedia.

PLATO achieved low precision for the Place-Area property because many places are ambiguous. For example, Athens can refer to either a city in Greece, Georgia, or Ohio. Similarly, Delaware can refer to either the U.S. state of Delaware or Delaware county in the U.S. state of Oklahoma. In the case of the later, given the entity pair of Delaware (State) and Oklahoma, PLATO may find false evidence supporting the hypothesis that the state of Delaware is part of Oklahoma, which can lead to poor precision. This problem can be addressed with richer partonomy semantics such as a state cannot be part of another state. These richer semantics are not captured by Winston's partonomy relationships (and hence the corresponding OWL properties), and offers a possible direction for future research.

Although we could not report recall, we provide preliminary insights into PLATO's performance on this measure. Our random sample of 100 entity pairs (see above) suggests PLATO achieved good performance on this metric. Of the 11 pairs with valid partonomy properties, PLATO discovered 7 of them. Moreover, qualitative observations of sample results further suggest that PLATO performs well on recall. For example, PLATO discovered the correct partonomy property between NATO and 23 of its member states – the total number of NATO member states is 28. Similarly, PLATO discovered the correct partonomy property between the Rock Band 'Nirvana' and all of its members – i.e. Kurt Cobain, Krist Novoselic and Dave Grohl.

The dataset and results used in this experiment are available at the project page[17], and we will continue to provide additional information related to partonomy as it becomes available.

4.2 Inter-Dataset Instance-Level Partonomy Discovery

We evaluated the performance of PLATO on discovering partonomy properties between entities from different LOD Cloud datasets using the following methodology. First, we created two inter-dataset partonomy discovery tasks: 1) discovering partonomy properties between Freebase dishes and

[14] such as [23]

[15] http://wiki.knoesis.org/index.php/PLATO

[16] http://en.wikipedia.org/w/api.php
?action=query&list=random&rnnamespace=0

[17] http://wiki.knoesis.org/index.php/PLATO

Relation Type	Distinct Entity Pairs	Correctly Found	Precision
Stuff-Object-Part-Of	4178	3427	0.82
Component-Integral-Part-Of	3126	27931	0.89
Feature-Activity-Part-Of	1287	464	0.85
Member-Collection-Part-Of	1912	803	0.85
Portion-Mass-Part-Of	0	0	NA
Place Area-Part-Of	3350	1248	0.48
Total	13853	10557	0.76

Table 2: Precision of the six different relation types between DBpedia entities

DBpedia ingredients, and 2) discovering partonomy properties between Freebase human anatomy parts and DBpedia organs. We chose these two tasks because:

- Freebase provides a pre-defined list of 2,615 food dishes[18] and 2,916 human anatomy parts,[19] which have well-defined parts (i.e. ingredient) and wholes (i.e. organ) respectively.
- DBpedia provides the corresponding parts and wholes.
- Freebase provides the ingredients for each food dish, which can be used as an independent gold standard for the first task; and experts in the medical domain were readily available to assess PLATO's performance for the second task.

We then applied PLATO to both tasks. For the Dish-Ingredient task, we validated the partonomy properties discovered by PLATO against the ingredients for each dish provided by Freebase to compute both precision (i.e. number of correct partonomy properties discovered by PLATO over all partonomy properties discovered) and recall (i.e. number of actual partonomy properties discovered by PLATO over all partonomy properties). For the Anatomy-Organ task, an independent gold standard does not exist – i.e. Freebase does not provide the organs for each anatomy part. Hence, we employed an expert in human anatomy to grade each partonomy property discovered by PLATO, and reported PLATO's precision based on the expert's response. These experts had no knowledge about PLATO and were presented the results as an exercise to judge if the presented ingredients are used for the given dish. The expert used the same grading criteria described in the previous experiment (see Section 4.1). We did not report the recall for PLATO because of resource and time limitations.

Task	Recall	Precision
Dish-Ingredient Task	0.72	0.53
Anatomy-Organ Task	N/A	0.86

Table 3: This table shows PLATO's performance on precision and recall for the Dish-Ingredient task, and PLATO's performance on precision for the Anatomy-Organ task. Recall was not reported for the second task because of time and resource limitations.

Table 3 shows the results for both tasks. For the Dish-Ingredient task, PLATO achieved high recall and modest precision. The Freebase dish gold standard consists of 2,615 dishes and a total of 1317 ingredients across these dishes. Many of the dishes do not have ingredients mentioned for them. PLATO discovered a total of 1766 partonomy relationships between Freebase dishes and DBpedia ingredients, of which 936 are valid according to the gold standard – giving a recall of 0.72 and precision of 0.53. This result demonstrates that PLATO can effectively discover partonomy properties across different LOD Cloud datasets. Interestingly, the modest precision was due to PLATO discovering additional, valid partonomy properties not present in the Freebase gold standard. For example, a stuff-object property exists between the ingredient ice cream and the dish 'Baked Alaska', which PLATO correctly discovered. However, the Freebase gold standard overlooked this relationship, resulting in lower precision.

Given this oversight, we employed 2 human graders to independently review each extra result generated (830 in total) to determine whether it's due to a real erroneous result given by PLATO or a gap in the gold standard (i.e. an overlooked ingredient in a food dish). The graders used the same grading criteria described in Section 4.1 We also required that both graders agree that a response is valid in order for it to be counted as correct. The graders responses were then used to adjust the precision. They found 512 correct answers out of 830, which resulted in total correct ingredients of 936+512=1448, an adjusted precision of 0.82 – a significant increase over the original precision.

For the Anatomy-Organ task, PLATO achieved high precision. Of the 8,397 distinct partonomy properties discovered by PLATO, the human expert verified 7,221 as correct, thus leading to a precision of 0.86. The expert in this case, is a researcher in medical science and not related to research and development of PLATO. The expert was presented the results of PLATO as a grading exercise to judge if the assertions are right or wrong. This result further demonstrates – in a different domain – that PLATO can effectively discover partonomy properties across different LOD Cloud datasets. For example, PLATO correctly identified that the entity 'Axon' is a component-integral object part of entities such as 'dorsal root ganglion', 'synapse', 'neuron' and 'nerve'. We plan to enrich Freebase's list of anatomy structures with the partonomy properties discovered by PLATO for this task.

4.3 Assertion of schema level links

Using the instance level assertions which are generated between entities, it becomes possible to identify the schema level relationships, which exist between the classes of these entities, as, described at the end of Section 3.2. For example, using the fact that 'Nirvana has a member Kurt Cobain' and

[18] http://www.freebase.com/view/food/views/dish

[19] http://www.freebase.com/view/medicine/views/anatomical_structure

'Queen has a member Freddie Mercury', and in fact that for all bands some member has been found which is classified as an artist, we are able to identify schema level assertions between DBpedia classes such as

```
dbpedia-owl:Band rdfs:subClassOf  [
   rdf:type            owl:Restriction ;
   owl:onProperty      :hasMember ;
   owl:someValuesFrom  dbpedia-owl:Artist
 ] .
```

The schema level statement essentially says that 'Bands have members Artists'. Table 4 shows the evaluation of precision for schema level links, which were asserted by PLATO.

Total # of Class Pairs	Correctly Identified	Precision
93	81	0.87

Table 4: Precision as measured on Schema Level Links Between DBpedia entities

The entity in column 1 in Table 4 is the total number of distinct class pairs that were asserted to have a relationship in the file expressing schema level constraints. For example [dbpedia-owl:Artist,dbpediaowl:Organization],[dbpedia-owl:Artist,dbpedia-owl:Artifact]. Thus, a single entity may occur in multiple such combinations, but in each of these pairs, the entity with which it is being related to is unique. Of these 93 different pairs, a total of 81 were found to be correct, leading to a precision of 0.87. The number of class pairs found is low because many entities in the DBpedia dataset do not have any classes associated with them. Identification of schema level relationships can potentially help with improving the precision and recall of instance level relationship identification. This dataset has also been made available on the project page for download.

5. RELATED WORK

To the best of our knowledge, this is the first work which, automatically identifies 'part-of' relationships in the context of the LOD cloud or RDF datasets. The field of Ontology Matching and Instance Matching has been focusing on identifying relationships such as 'sameAs','subClass' and 'equivalentClass.' In [10, 8] the authors present a survey in the area of ontology matching. This helps in cleaning up the data and improving the quality of links at the instance level, but the issue of identifying appropriate relationships at the schema level has not been addressed. voiD [1] provides a vocabulary to represent the relationships between the different datasets. SILK Framework [32] automates the process of link discovery between LOD datasets at the instance level. At the schema level, a notable effort for creating a unified reference point for LOD schemas is UMBEL [3], which is a coherent framework for ontology development and can serve as a reference framework.

There has been a number of efforts in the area of Natural Language Processing for identification of part-of relationships within a text corpora [14, 30]. This includes effort that utilizes the presence of certain lexico-syntactic patterns (Hearst patterns [16]) to indicate a particular semantic relationship between two nouns. However, much of this work has been confined to ontology learning [9] in the sense of hyponym extraction [16]. A closely related work that also mines the Web for the relations is NELL [6]. There are a few notable differences between our approach and NELL, (1) NELL uses a crawler to crawl the Web and identify relations it can find between entities on the web. We are focused on LOD cloud and for a given pair of entities, PLATO tries to identify the relationship between them. (2) Predicates or properties extracted from NELL are at the surface level and do not convey the semantics of the properties. For example, while NELL does extracts fact such as Athens and Greece are related by the predicate citycapitalofcountry, it does not explicitly provides any semantics to those relationships. We have definitely gained a lot of insight from the work of NELL and it also validates our belief that web can be mined to gain information about relationships. However, it will be extremely difficult to compare PLATO with NELL since, NELL is not available for download and systems have different set up and objectives.

The closest work in this respect is Espresso [26] that again works on a specific text corpus. A key difference of this work from ours is its use of a supervised approach. Further, it disregards any information about the type of entities, which we capture using Winston's patterns.

6. POTENTIAL IMPACT & FUTURE WORK

To the best of our knowledge, this is the only work that can identify partonomic relations between entities in the LOD Cloud. The potential impact of this work is many fold in the context of the LOD Cloud and beyond. Our work suggests that introducing the part-of relationship as a standard ingredient in and between LOD Cloud datasets is viable. This will allow LOD to move beyond the sameAs relationship and allow it to be used for more meaningful purposes. The discovery of individual components of various entities such as body parts or organizations may enable the identification of new scientific facts and the answering of analytical queries. The extension of Freebase to incorporate this information for dishes and human anatomy is something we would like to address in the short term. We would also like to add partonomical relations between entities of other LOD datasets. The additional schema information generated by PLATO will also be made available as a part of the LOD cloud for use by the reasoning community. The low precision on the Place-Area relationship is a matter of concern and we plan to address it in near future. We would also like to evaluate the results for Anatomy-Organ Task using a domain specific ontology such as Foundational Model of Anatomy [20].

We plan on contributing the entire corpus of entities that have been identified to be in part-of relationship as a dataset to the LOD cloud. This will prove useful for researchers who wish to utilize the dataset and also for any comparative evaluation in the future. We have done an initial testing of our approach on identification of other relationships such as 'causality' and it appears promising. We would like to extend it further and develop techniques for the identification of these relations, eventually leading to a rich Relationship Web. There is also plenty of scope for the improvement of our own technique as well. We would like to be able to extend PLATO to identify fundamental relationships. We

[20] http://fma.biostr.washington.edu/

would like to further strengthen the schema learning part by adding established inductive methods. We would also like to add additional capabilities for entity disambiguation to improve precision and recall figures. We would also like to explore the use of schema knowledge generated by PLATO to improve instance matching, leading to a system with a feedback loop.

7. CONCLUSION

In this paper we have presented an automatic approach (PLATO) for identification of part-of relation between entities in the LOD cloud. These entities can be part of the same dataset or can belong to different datasets. In addition, the entities can be either instances or classes. Our approach is based on the foundational work by Winston in the area of partonomy and the corresponding taxonomy for the same. Since Winston's work is more tailored towards linguistics, we have expressed the work using OWL constraints in order to operationalize it for the purpose of our work. We described the technical solution used to provide PLATO and also presented a comprehensive evaluation spanning thousands of entities in the LOD cloud. Our results demonstrate that PLATO identifies part-of relationships between entities in the LOD cloud with a fairly high precision.

We believe our solution works well because of the following reasons (1) We utilize a rich datasource 'the Web' to identify the relationship between entities (2) Our approach has a foundational underpinning on a classical work in partonomical relation.

8. ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under award 1143717 "III: EAGER – Expressive Scalable Querying over Linked Open Data."

9. REFERENCES

[1] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing Linked Datasets – On the Design and Usage of voiD, the 'Vocabulary of Interlinked Datasets'. In *WWW2009 Workshop on Linked Data on the Web (LDOW2009)*, Madrid, Spain, 2009.

[2] A. Artale, E. Franconi, N. Guarino, and L. Pazzi. Part-whole relations in object-centered systems: An overview. *Data & Knowledge Engineering*, 20(3):347–383, 1996.

[3] Michael K. Bergman and Frédérick Giasson. UMBEL ontology, volume 1, technical documentation. Technical Report 1, Structured Dynamics, 2008. Available from: http://umbel.org/doc/UMBELOntology_vA1.pdf.

[4] Christian Bizer, Tom Heath, and Tim Berners Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

[5] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia—A crystallization point for the Web of Data. *Journal of Web Semantics*, 7(3):154–165, 2009.

[6] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*, 2010.

[7] R. Casati and A.C. Varzi. *Parts and places: The structures of spatial representation*. The MIT Press, 1999.

[8] Namyoun Choi, Il-Yeol Song, and Hyoil Han. A survey on ontology mapping. *SIGMOD Rec.*, 35(3):34–41, 2006.

[9] Philipp Cimiano, Andreas Hotho, and Steffen Staab. Learning concept hierarchies from text corpora using formal concept analysis. *J. Artif. Int. Res.*, 24:305–339, August 2005.

[10] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.

[11] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, illustrated edition edition, May 1998.

[12] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, and John Prager. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79, 2010.

[13] P. Gerstl and S. Pribbenow. A conceptual theory of part-whole relations and its applications. *Data & Knowledge Engineering*, 20(3):305–322, 1996.

[14] R. Girju, A. Badulescu, and D. Moldovan. Automatic discovery of part-whole relations. *Computational Linguistics*, 32(1):83–135, 2006.

[15] Michael Hausenblas. Exploiting linked data to build web applications. *IEEE Internet Computing*, 13:68–73, 2009.

[16] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics – Volume 2*, COLING '92, pages 539–545, Stroudsburg, PA, USA, 1992.

[17] P. Hitzler, M. Krötzsch, B. Parsia, P.F. Patel-Schneider, and S. Rudolph, editors. *OWL 2 Web Ontology Language: Primer*. W3C Recommendation, 27 October 2009. Available at http://www.w3.org/TR/owl2-primer/.

[18] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. W3C Member Submission 21 May 2004, 2004. Available from http://www.w3.org/Submission/SWRL/.

[19] Prateek Jain, Pascal Hitzler, Peter Z. Yeh, Kunal Verma, and Amit P. Sheth. Linked Data is Merely More Data. In *Linked Data Meets Artificial Intelligence*, pages 82–86. AAAI Press, Menlo Park, CA, 2010.

[20] Prateek Jain, Peter Z. Yeh, Kunal Verma, Cory A. Henson, and Amit P. Sheth. SPARQL query re-writing using partonomy based transformation rules. In *Proceedings of the 3rd International Conference on GeoSpatial Semantics*, GeoS '09, pages 140–158, Berlin, Heidelberg, 2009. Springer-Verlag.

[21] Adila Krisnadhi, Frederick Maier, and Pascal Hitzler.

OWL and Rules. In *Reasoning Web. Semantic Technologies for the Web of Data – 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures*, volume 6848 of *Lecture Notes in Computer Science*, pages 382–415. Springer, Heidelberg, 2011.

[22] Markus Krötzsch, Frederick Maier, Adila A. Krisnadhi, and Pascal Hitzler. A better uncle for OWL: Nominal schemas for integrating rules and ontologies. In *Proceedings of the 20th International World Wide Web Conference, WWW2011, Hyderabad, India, March/April 2011*, pages 645–654. ACM, New York, 2011.

[23] Jens Lehmann and Pascal Hitzler. Concept learning in description logics using refinement operators. *Machine Learning*, 78(1–2):203–250, 2010.

[24] B. Motik, P.F. Patel-Schneider, and B. Parsia, editors. *OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax*. W3C Recommendation, 27 October 2009. Available at http://www.w3.org/TR/owl2-syntax/.

[25] Boris Motik, Ulrike Sattler, and Rudi Studer. Query answering for OWL DL with rules. *Journal of Web Semantics*, 3(1):41–60, 2005.

[26] Patrick Pantel and Marco Pennacchiotti. Espresso: leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 113–120, Stroudsburg, PA, USA, 2006.

[27] Alan Rector, Chris Welty, Natasha Noy, and Evan Wallace. Simple part-whole relations in OWL Ontologies available at http://www.w3.org/2001/sw/bestpractices/oep/simplepartwhole/, August 2005.

[28] Barry Smith. The basic tools of formal ontology. In *Formal Ontology in Information Systems*, 1998.

[29] Nectaria Tryfona and Max J. Egenhofer. Consistency among parts and aggregates: A computational model. *Transactions in GIS*, 1(3):189–206, 1996.

[30] Willem van Hage, Hap Kolb, and Guus Schreiber. A method for learning part-whole relations. In *The Semantic Web - ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 723–735. Springer Berlin / Heidelberg, 2006.

[31] J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Silk–A Link Discovery Framework for the Web of Data. In *2nd Linked Data on the Web Workshop (LDOW2009)*, Madrid, Spain, 2009. Available from http://ceur-ws.org/Vol-538/ldow2009_paper13.pdf.

[32] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Discovering and maintaining links on the web of data. In *ISWC '09: Proceedings of the 8th International Semantic Web Conference*, pages 650–665, Berlin, Heidelberg, 2009. Springer-Verlag.

[33] Morton E. Winston, Roger Chaffin, and Douglas Herrmann. A taxonomy of part-whole relations. *Cognitive Science*, 11(4):417–444, 1987.