Provision and Usage of Provenance Data in the WebIsALOD Knowledge Graph

Sven Hertling and Heiko Paulheim

Data and Web Science Group, University of Mannheim, Germany
{sven,heiko}@informatik.uni-mannheim.de

Abstract. The WebIsALOD dataset provides a linked data endpoint to the WebIsA database, which harvests millions of subsumption relations from a large scale Web crawl using text patterns. For each of the relations, the dataset also contains rich provenance data, such as the text pattern used, the original sentence in which the pattern was found, and the source on the Web. In this paper, we describe several alternatives and design decisions for providing statement-level provenance information at large scale for the WebIsALOD dataset. Furthermore, we show the practical impact of that provenance information for computing confidence scores approximating the correctness of each subsumption relation.

Keywords: Provenance, Knowledge Graph, Reification, Singleton Property, Named Graph

1 The WebIsALOD Knowledge Graph

WebIsALOD is a large-scale, cross-domain Semantic Web Knowledge Graph, which provides subsumption relations between entities recognized on the Web. The knowledge graph has been created from an initial extraction of this information, i.e., the WebIsADB dataset [15], in order to provide a service in line with Linked Data standards and best practices [14].

The main idea of the WebIsADB is to extract hypernymy relations from a huge and fixed web crawl called CommonCrawl[1]. The extraction method is based on 58 Hearst-like lexico-syntactic patterns [4] which are frequent patterns to describe hypernymy relations. For example, the sentence *Still, people use Gmail and other Web services* implies the hypernymy relation between *Gmail* and *Web service*, which can be captured by the pattern *NP and other NP*.[2] The original dataset contains 400,533,808 relations, 120,992,255 unique hyponyms, and 107,691,822 unique hypernyms. Thus, the knowledge graph contains many more instances than the popular public knowledge graphs such as DBpedia [13].

For providing the WebIsADB as Linked Data, we represent the hypernymy relations using SKOS[3] via the `skos:broader` relation. As described in [6], the

[1] `https://commoncrawl.org`

[2] *NP* stands for *noun phrase*.

[3] `https://www.w3.org/TR/skos-reference/`

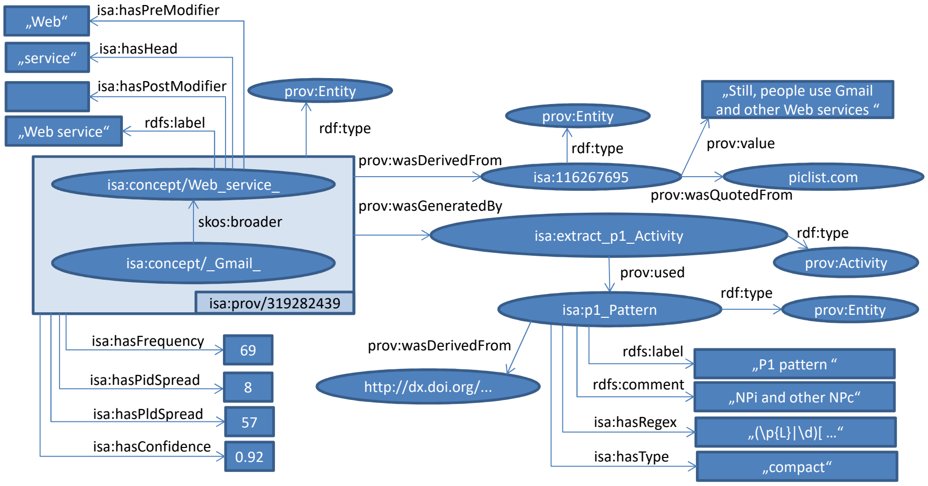


Fig. 1: Example for a subsumption relation in the WebsALOD dataset, including provenance information

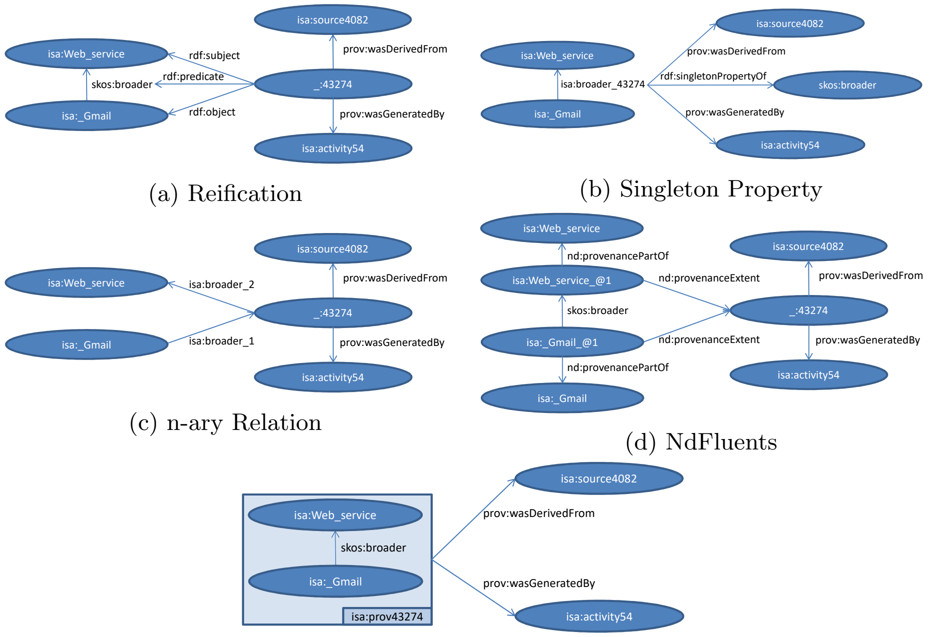
original dataset contains a lot of noisy extractions, therefore, we train a machine learning model to compute a confidence score for each relation (see section 4). The final resulting dataset consists of the original 400,533,808 hypernymy relations, together with a confidence score and provenance metadata (see below), as well as 2,593,181 instance links to DBpedia [7] and 23,771 class links to YAGO [16]. All in all, the dataset consists of 5.4B triples [6]. The dataset is available online as a Linked Data service, a SPARQL endpoint, and as an RDF dump.[4]

2 Provenance Information Provided

For each single relation, the WebIsADB also collects the information how it was created – i.e., the originating sentence, its source, and the pattern that was used to find the relation. Furthermore, statistical metadata is computed from that information, i.e., the overall number of observations, the number of different patterns and the number of different sources in which the relation was found. Additionally, we include a pointer to a scientific literature source for each pattern (i.e., the paper in which the pattern was proposed). Where possible, we reused constructs from the PROV ontology[5], while we created our own properties where no suitable concepts were defined in that ontology

For each entity (i.e., hypernym or hyponym), we also provide information generated during the syntactic analysis which is performed to extract the statement, i.e., the head noun and potential pre and post modifiers. The big picture

[4] http://webisa.webdatacommons.org/
[5] https://www.w3.org/TR/prov-o/



(a) Reification

(b) Singleton Property

(c) n-ary Relation

(d) NdFluents

(e) Named Graphs

Fig. 2: Schematic Depiction of Alternatives for Providing Metadata

is shown in Fig. 1. Note that for each relation, multiple sources and patterns can be provided.

3 Alternatives for Providing Provenance Metadata

In the WebIsALOD knowledge graph, we provide provenance information on statement level, i.e., for each single triple with a `skos:broader` relation, metadata has to e attached to that very triple. To that end, we explored different alternatives, which are shown in Fig. 2. Each of them has its own advantages and disadvantages.

3.1 RDF Reification

RDF provides a means called reification to make statements about statements. For each statement to be reified, a single RDF node representing the triple is created, which has a relation to the subject, the predicate, and the object.

On the positive side, RDF reification is well understood, since it is rather intuitive and covered in many Semantic Web documentations, tutorials, and text books[6]. On the negative side, the number or RDF triples is drastically increased – a single triple has to be replaced by four triples to allow for reification.

[6] e.g., the W3C RDF primer, `https://www.w3.org/TR/rdf-primer/`

```
_:43274 a rdf:Statement .
_:43274 rdf:subject isa:_GMail_ .
_:43274 rdf:predicate skos:broader .
_:43274 rdf:object isa:Web_Service_ .
_:43274 prov:wasDerivedFrom isa:source4082 .
_:43274 prov:wasGeneratedBy isa:activity54 .
```

(a) Knowledge Graph

```
SELECT DISTINCT ?label WHERE {
        ?s1 a rdf:Statement ;
            rdf:subject isa:_GMail_ ;
            rdf:predicate skos:broader ;
            rdf:object ?x .
        ?s2 a rdf:Statement ;
            rdf:subject ?x ;
            rdf:predicate skos:broader ;
            rdf:object ?y .
        ?y rdfs:label ?label .
        ?s1 isaont:hasConfidence ?c1 .
        ?s2 isaont:hasConfidence ?c2 .
        FILTER(?c1>0.75 && ?c2>0.75)
}
```

(b) SPARQL Query

Fig. 3: RDF Reification

In our case, this would mean that to represent 400M subsumption relations, 1.6B RDF triples would be required for the statements alone, not including any provenance information. Likewise, SPARQL queries against such a dataset involving both the subsumptions as well as the provenance information can become rather complex.

In the following, we will use the example of querying for ancestors of a fixed concept which are two levels up (i.e., broader terms of broader terms of a fixed concept), and both subsumption relations are required to have a minimum confidence of 0.75. Using reification, this query would look as in figure 3b.

3.2 Singleton Properties

An alternative proposed in [10] is to define a singleton property for each subsumption relation. This property can be made an instance of the desired relation (in our case: `skos:broader`) and is then used as a subject of the attached provenance information. This approach is slightly less verbose than RDF reification.

On the downside, in the case of WebIsALOD, the resulting schema with 400M direct subproperties of `skos:broader` could be regarded as slightly deteriorated, and there are experience reports with large-scale knowledge graph that hint at some triple stores suffering from such large numbers of singleton properties [5]. Moreover, with this approach, additional important properties of the `skos:broader` relation, in particular, transitivity, has to be taken particular care of when implementing the semantics of `rdf:singletonPropertyOf`.

The above query reformulated with singleton properties is shown in figure 4b.

3.3 n-ary Relations

While RDF in its native form only supports binary relations, a pattern for the representation of n-ary relations has been proposed as well [11], naming the utilization for representing context information of a relation as a possible use

```
isa:_GMail_ isa:broader_43274 isa:Web_Service_ .
isa:broader_43274 rdf:singletonPropertyOf skos:broader .
isa:broader_43274 prov:wasDerivedFrom isa:source4082.
isa:broader_43274 prov:wasGeneratedBy isa:activity54 .
```

(a) Knowledge Graph

```
SELECT DISTINCT ?label WHERE {
        isa:_GMail_ ?p1 ?x .
        ?x ?p2 ?y .
        ?p1 rdf:singletonPropertyOf skos:broader .
        ?p2 rdf:singletonPropertyOf skos:broader .
        ?y rdfs:label ?label .
        ?p1 isaont:hasConfidence ?c1 .
        ?p2 isaont:hasConfidence ?c2 .
        FILTER(?c1>0.75 && ?c2>0.75)
}
```

(b) SPARQL Query

Fig. 4: Singleton Properties

```
isa:_GMail_ isa:broader_1 _:43274 .
_:43274 isa:broader_2 isa:Web_Service_ .
_:43274 prov:wasDerivedFrom isa:source4082 .
_:43274 prov:wasGeneratedBy isa:activity54 .
```

(a) Knowledge Graph

```
SELECT DISTINCT ?label WHERE {
        isa:_GMail_ isa:broader_1 ?r1 .
        ?r1 isa:broader_2 ?x .
        ?x isa:broader_1 ?r2 .
        ?r2 isa:broader_2 ?y .
        ?y rdfs:label ?label .
        ?r1 isaont:hasConfidence ?c1 .
        ?r2 isaont:hasConfidence ?c2 .
        FILTER(?c1>0.75 && ?c2>0.75)
}
```

(b) SPARQL Query

Fig. 5: n-ary Relations

case. The relation itself is represented as a blank node here, and the original relation is broken down into two.

The verbosity of this variant is fairly low, requiring only the blank node for the relation as an additional resource. Transitivity of the `skos:broader` relation can also be built into the model by using OWL 2 property chains, i.e., exploiting the transitivity of `skos:broader` would require an OWL 2 reasoner, compared to standard OWL Lite reasoning for exploiting the simple transitivity of the original definition. Moreover, in the original description of the pattern, OWL constraints for the new relations are also defined, using both universal and existential quantification, and hence leaving even the tractable OWL EL fragment.

Using n-ary relations, the above query would look like in figure 5b.

3.4 NdFluents

NdFluents is an ontology and a set of design patterns proposed in [3]. It is an extension of the 4dFluents ontology for adding temporal context to statements without changing the RDF data model [17], which it extends to arbitrary context information beyond temporal context. The authors argue that this approach is better suited for preserving inference than using RDF reification.

The NdFluents approach foresees the creation of a “copy” for both the subject and the object to attach context information to.

We can observe that the number of triples is even larger than for reification. Moreover, the new resources need to be created for each single statement

```
isa:_GMail_@1 skos:broader isa:Web_Service_@1 .
isa:_GMail_@1 nd:provenancePartOf isa:_GMail_ .
isa:Web_Service_@1 nd:provenancePartOf isa:_GMail_ .
isa:_GMail_@1 nd:provenanceExtent _:43274 .
isa:Web_Service_@1 nd:provenanceExtent _:43274 .
_:43274 prov:wasDerivedFrom isa:source4082 .
_:43274 prov:wasGeneratedBy isa:activity54 .
```

(a) Knowledge Graph

```
SELECT DISTINCT ?label WHERE {
        ?gmail1 nd:provenancePartOf isa:_GMail_ .
        ?gmail1 skos:broader ?x1 .
        ?x1 nd:provenancePartOf ?x.
        ?x2 nd:provenancePartOf ?x .
        ?x2 skos:broader ?y1 .
        ?y1 nd:provenancePartOf ?y .
        ?y rdfs:label ?label .
        ?x1 nd:provenanceExtent ?e1 .
        ?x2 nd:provenanceExtent ?e2 .
        ?x1 isaont:hasConfidence ?c1 .
        ?x2 isaont:hasConfidence ?c2 .
        FILTER(?c1>0.75 && ?c2>0.75)
}
```

(b) SPARQL Query

Fig. 6: NdFluents

a resource is involved in. For example, the resource `isa:_president_` has 1,821 hyponyms and 4,656 hypernyms, which would require the creation of 6,477 new resources alone for representing the resource `isa:_president_`. In total, for WebIsALOD, 400M relations would require the creation of 800M additional resources, i.e., increasing the number of resources in the dataset by a factor of more than four.

The query above, formulated against an NdFluents dataset, is shown in figure 6b.

3.5 RDF Graphs

RDF named graphs form collections of RDF statements, which are said to belong to a certain graph. Such a collection of RDF statements in an RDF graph is assigned a URI (which makes it a *named* graph) and can be used as a subject and/or object of other statements [1]. Often, RDF named graphs are represented using RDF quads.[7] For WebIsALOD, we turn every subsumption into its own named graph, which is then used as a subject of further provenance information (in the WebIsALOD main graph).

Like RDF reification, named graphs are also easily understood, and the RDF quad notation allows for relatively simple formulation of statements and efficient SPARQL queries. Furthermore, the use of RDF reification is often discouraged in the Linked Data context in favor of using graphs and quads instead [9]. On the downside, RDF graphs are originally meant to hold a *collection* of RDF triples, whereas creating a single named graph for each triple, as in our case, can be regarded as a slightly abusive use of named graphs.

The query example would look like in figure 7b.

3.6 Design Decision

Looking at the considerations above may lead to different conclusions, depending on which criteria are deemed more important. Our aim was to provide a

[7] `https://www.w3.org/TR/n-quads/`

```
isa:_GMail_ skos:broader isa:Web_Service_ isa:prov43274.
isa:prov43274 prov:wasDerivedFrom isa:source4082 .
isa:prov43274 prov:wasGeneratedBy isa:activity54 .
```

(a) Knowledge Graph

```
SELECT DISTINCT ?label WHERE{
        GRAPH ?g1 {
                isa:_GMail_ skos:broader ?x .
        }
        GRAPH ?g2 {
                ?x skos:broader ?y .
        }
        ?y rdfs:label ?label.
        ?g1 isaont:hasConfidence ?c1.
        ?g2 isaont:hasConfidence ?c2.
        FILTER(?c1>0.75 && ?c2>0.75)
}
```

(b) SPARQL Query

Fig. 7: RDF Graphs

dataset which is versatile enough to satisfy different use cases, as well as allows good usability and understandability to ease adoption as much as possible. Additionally, given the sheer data volume, the verbosity should not be too high, i.e., not multiply the original dataset's size by a larger factor. Another important aim was to allow exploitation of the transitivity of the *skos:broader*, i.e., easily retrieving *all* hyponyms or hypernyms of a concept.

Apart from those theoretic aspects, practical considerations also played a role in the design decision. Due to the sheer volume of the dataset, we had to pick an RDF triple store which can handle such a large knowledge graph, therefore, we chose Virtuoso [2], which is free software and at the same time has been shown to be highly scalable [8]. Consulting the documentation, we found that Virtuoso also recommends the use of Named Graphs, whereas the documentation states that "the RDF reification vocabulary can be used [...] It is however very inefficient and is not supported by any specific optimization."[8] Therefore, RDF Named Graphs were ultimately used to implement provenance information in the WebIsALOD knowledge graph.

4 Exploitation of Provenance Information

Since the extraction of the original WebIsADB dataset was focused on coverage rather than correctness, it contains quite a few noisy extractions. Hence, we had to apply some post refinement of the knowledge graph to be able to serve a dataset which as a useful quality [12]. Instead of filtering statements, we have decided to follow the spirit of the original dataset, i.e., not reducing the coverage, but to rather provide confidence values for each statement. That way, consumers of the dataset can control the trade off between coverage and correcntess themselves, depending on the use case at hand. At the same time, the confidence scores are also used to order the results in the dataset's front end, showing the most trusted statements at the top.

As shown in [6], rating statements only by frequency is not a good indicator of quality. Basically, each statement observed with more than one pattern and

[8] `https://www.openlinksw.com/weblog/oerling/?id=1572`

on more than one source has the same likelihood of being correct, regardless of the actual frequency. At the same time, this likelihood is fairly low (below 35%), which makes this approach not suitable for curating a dataset of high quality.

On the other hand, the information contained in the provenance metadata can be a useful indicator for rating the correctness of a statement: e.g., some patterns may be prone to creating more noise than others, and a larger spread of patterns and sources may be a better indicator for statement correctness.

For the WebIsALOD dataset, we trained a machine learning model to capture such meta-patterns and exploited it to rate the correctness of all statements in the dataset. More precisely, we had a ground truth dataset annotated by means of crowd sourcing, indicating the correctness or incorrectness of a statement. This dataset was then used to train a classifier to tell correct from incorrect statements, and the confidence score provided by the classifier is added to the provenance data as a confidence score of the statement. A RandomForest classifier has been shown to achieve an area under the ROC curve of up to .84, i.e., it can assign rather precise confidence scores. [6]

Using those scores, it is possible to set a threshold for the quality of the relations when querying the knowledge graph, as in the examples above.

5 Conclusion

In this paper, we have shown how provenance information is used in the WebIsALOD knowledge graph. The dataset contains a large volume of provenance metadata, which is attached to individual statements.

Adding statement level provenance information to a dataset of that size does not come without challenges. We have explored different alternatives and decided to use named graphs for providing provenance information. However, this is a decision that we deemed suitable for the knowledge graph at hand, and other datasets with other characteristics (e.g., different sizes, larger number of statements sharing the same provenance information), and/or another underlying tool stack, might be better suited using other approaches.

We hope that this paper can inspire other dataset providers to add fine-grained provenance information, since provenance information is still not used by the majority of datasets on the LOD cloud [14], and that the experience shared in this paper might serve as helpful advice for implementing provenance in a way that suits the dataset at hand.

References

1. Carroll, J.J., Bizer, C., Hayes, P., Stickler, P.: Named graphs. Web Semantics: Science, Services and Agents on the World Wide Web 3(4), 247–267 (2005)
2. Erling, O.: Virtuoso, a hybrid rdbms/graph column store. IEEE Data Eng. Bull. 35(1), 3–8 (2012)
3. Giménez-García, J.M., Zimmermann, A., Maret, P.: Ndfluents: An ontology for annotated statements with inference preservation. In: European Semantic Web Conference. pp. 638–654. Springer (2017)

4. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of COLING '92. pp. 539–545 (1992)
5. Hernández, D., Hogan, A., Krötzsch, M.: Reifying rdf: What works well with wikidata? SSWS ISWC 1457, 32–47 (2015)
6. Hertling, S., Paulheim, H.: Webisalod: providing hypernymy relations extracted from the web as linked open data. In: ISWC. pp. 111–119 (2017)
7. 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. Semantic Web Journal 6(2) (2013)
8. Morsey, M., Lehmann, J., Auer, S., Ngomo, A.C.N.: Dbpedia sparql benchmark–performance assessment with real queries on real data. In: ISWC. pp. 454–469 (2011)
9. Ngomo, A.C.N., Auer, S., Lehmann, J., Zaveri, A.: Introduction to linked data and its lifecycle on the web. In: Reasoning Web International Summer School. pp. 1–99 (2014)
10. Nguyen, V., Bodenreider, O., Sheth, A.: Don't like rdf reification?: making statements about statements using singleton property. In: Proceedings of the 23rd international conference on World wide web. pp. 759–770. ACM (2014)
11. Noy, N., Rector, A.: Defining n-ary relations on the semantic web (2006), https://www.w3.org/TR/swbp-n-aryRelations/
12. Paulheim, H.: Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods. Semantic Web (2016)
13. Ringler, D., Paulheim, H.: One knowledge graph to rule them all? analyzing the differences between dbpedia, yago, wikidata & co. In: 40th German Conference on Artificial Intelligence (2017)
14. Schmachtenberg, M., Bizer, C., Paulheim, H.: Adoption of the linked data best practices in different topical domains. In: ISWC. pp. 245–260 (2014)
15. Seitner, J., Bizer, C., Eckert, K., Faralli, S., Meusel, R., Paulheim, H., Ponzetto, S.P.: A large database of hypernymy relations extracted from the web. In: LREC (2016)
16. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. In: 16th international conference on World Wide Web. pp. 697–706 (2007)
17. Welty, C., Fikes, R., Makarios, S.: A reusable ontology for fluents in owl. In: FOIS. vol. 150, pp. 226–236 (2006)