Data-Driven RDF Property Semantic-Equivalence Detection using NLP Techniques

Mariano Rico, Nandana Mihindukulasooriya, and Asunción Gómez-Pérez

Ontology Engineering Group, Universidad Politécnica de Madrid, Spain
{mariano.rico,nmhinidu,asun}@fi.upm.es

Abstract. DBpedia extracts most of its data from Wikipedia's infoboxes. Manually-created "mappings" link infobox attributes to DBpedia ontology properties (dbo properties) producing most used DBpedia triples. However, infoxbox attributes without a mapping produce triples with properties in a different namespace (dbp properties). In this position paper we point out that (a) the number of triples containing dbp properties is significant compared to triples containing dbo properties for the DBpedia instances analyzed, (b) the SPARQL queries made by users barely use both dbp and dbo properties simultaneously, (c) as an exploitation example we show a method to automatically enhance SPARQL queries by using syntactic and semantic similarities between dbo properties and dbp properties.

Keywords: SPARQL query; query enhancement; DBpedia; Spanish DBpedia; property mapping

1 Introduction

DBpedia [1] is the central hub of the Linked Open Data (LOD) cloud because it provides a vast amount of information and most of the datasets in the LOD cloud link to DBpedia. The extraction process [2] in DBpedia generates properties of two types: (1) properties in the DBpedia ontology (we name these dbo properties), and (2) properties not in the DBpedia ontology (let us name them dbp properties). The dbp properties come from the attribute-value pairs found in Wikipedia infoboxes that has no manually-created mappings [1]. The analysis of the Spanish DBpedia (esDBpedia) found [3] that, despite the high number of mappings (100+ classes), for each 4 triples containing a dbo property there is 1 triple containing a dbp property. In this work, we extend this analysis to English and German DBpedia instances, with similar results. For instance, in the English DBpedia this ratio goes to almost one to one.

In this position paper we hypothesize that triples can not be accessed because most queries are comprised of dbo properties. DBpedia defines around 2500 properties, but only 2% infoboxes fields are mapped to the DBpedia ontology.

[1] See DBpedia multilingual mappings at http://mappings.dbpedia.org

Thus, there are many dbp properties in DBpedia: 58,239 for the English version, 17,111 for the Spanish and 12,167 for the German. Therefore, users that query the DBpedia endpoint by using SPARQL queries containing only dbo properties have no access to a significant amount of triples and could lead to null or incomplete results even if the relevant data is available in DBpedia.

In this work, we start by checking the assumption that users barely mix dbp and dbo properties in SPARQL queries. Later we provide a method to automatically identify the most similar dbp properties for a given dbo property. This method takes advantage of techniques from Natural Language Processing and Statistical Methods. The goal of the proposed method is to generate "automatic mappings" with a certain confidence level. These mappings can be manually approved by a specialist or through crowd-sourcing in a semi-automatic manner. Some examples point out that these mappings can enhance the SPARQL queries to generate better results by accessing more information in different DBpedia instances.

2 Background

In this section, we explore two hypotheses addressed in this paper. On the one hand, we analyze the amount of information described by using dbp properties in 3 DBpedia instances. On the other hand, we analyze how dbo and dbp properties are used in SPARQL queries made to the English DBpedia.

Firstly, table 1 shows for three DBpedia instances (English, Spanish and German) the following data: the number of dbo and dbp properties, the number of triples containing those properties, and the top-10 dbp properties ordered by the number of triples containing those properties. The ratio dbp/dbo (number of triples with dbp properties per number of triples with dbo properties) goes to 0.95, 0.32, and 0.20 respectively. That is, the English DBpedia has the highest ratio, with almost as much triples containing dbo properties as triples containing dbp properties.

Secondly, we analyzed a SPARQL query log to evaluate the assumption that users do not frequently use dbp properties in their SPARQL queries. We used the Linked SPARQL Queries Dataset [4], which provides a RDF model to know details about SPARQL queries made to several endpoints. We explored the data from the English DBpedia to see how many queries use both dbp and dbo properties. Out of 1,208,762 distinct queries only 2,328 queries use both dbo and dbp properties in the same query. We made a similar analysis for agents (IPs): out of 3,041 distinct agents (IPs), only 473 use both dbp and dbo properties in the same SPARQL query. This illustrates that the majority of the SPARQL queries miss some portion of the data. We argue that this information can be reached by enhancing the SPARQL queries by using our proposed mappings between dbo and dbp properties.

Table 1. Top-10 dbp properties for the English, Spanish and German DBpedia instances (2015-04 version).

English DBpedia		**Spanish DBpedia**		**German DBpedia**	
dbp: http://dbpedia.org/property/					
URI	Triples	URI	Triples	URI	Triples
dbp:hasPhotoCollection	4,041,585	dbp:wikiPageUsesTemplate	3,402,499	dbp:name	494,852
dbp:name	4,021,368	dbp:nombre	558,837	dbp:geburtsort	305,063
dbp:title	1,452,504	dbp:título	327,498	dbp:kurzbeschreibung	283,695
dbp:subdivisionType	1,257,766	dbp:name	230,763	dbp:geburtsdatum	283,405
dbp:shortDescription	1,194,274	dbp:tipoSuperior	225,868	dbp:typ	232,702
dbp:dateOfBirth	1,023,951	dbp:horario	203,890	dbp:viaf	169,145
dbp:subdivisionName	1,004,294	dbp:imagen	183,887	dbp:gnd	165,362
dbp:goals	969,216	dbp:familia	152,430	dbp:jahre	156,498
dbp:placeOfBirth	908,819	dbp:title	144,724	dbp:sterbedatum	144,209
dbp:birthPlace	903,529	dbp:ordo	142,196	dbp:alternativnamen	143,893
#props dbp	58,239	#props dbp	17,111	#props dbp	12,167
#props dbo	1,338	#props dbo	559	#props dbo	534
#triples dbp	78,125,087	#triples dbp	28,234,292	#triples dbp	10,483,987
#triples dbo	82,369,408	#triples dbo	90,389,560	#triples dbo	50,750,486

3 An approach for automatically enhancing SPARQL queries

Figure 1 shows, from top to down, the process for finding 'similar' dbp properties for a given dbo property. The first step (figure top side) is to aggregate properties into groups according to their domain and range. The objective of this grouping is to work with smaller groups of properties with potentially similar semantics. For dbo properties, domain and range are specified by the DBpedia ontology, but dbp properties have no explicit domain or range. However, we can estimate domain and range by using tools such as LOUPE [5] (`http://loupe.linkeddata.es`) which provides domain and range for dbp properties analyzing the subject and the object of all triples containing a given dbp property. Following the figure, after this aggregation, properties that have *dbo:Person* as domain and *dbo:Place* as range are located in a smaller group which includes, among others, dbp properties like *dbp:birthPlace*, *dbp:birzPlace* and *dbp:deathPlace*, as well as dbo properties like *dbo:birthPlace* or *dbo:birthLocation*.

The second step involves processing each small group by using Natural Language pre-Processing which includes tokenization and stemming/lemmatization. Many dbp properties are compound words (e.g. birthPlace → (birth, place)). It is necessary to do some pre-processing for tokenizing those properties before applying linguistic techniques to find syntactic and semantic similarity. For dbp properties that use the camel case convention, this tokenization can be done easily by breaking the words using the camel case convention. For the rest, for instance the dbp properties that use all simple letters (e.g. *oldcode* or *testaverage*) or all capitals, dictionary tools that break the compound words into separate tokens of known words can be used. We also used other punctuation

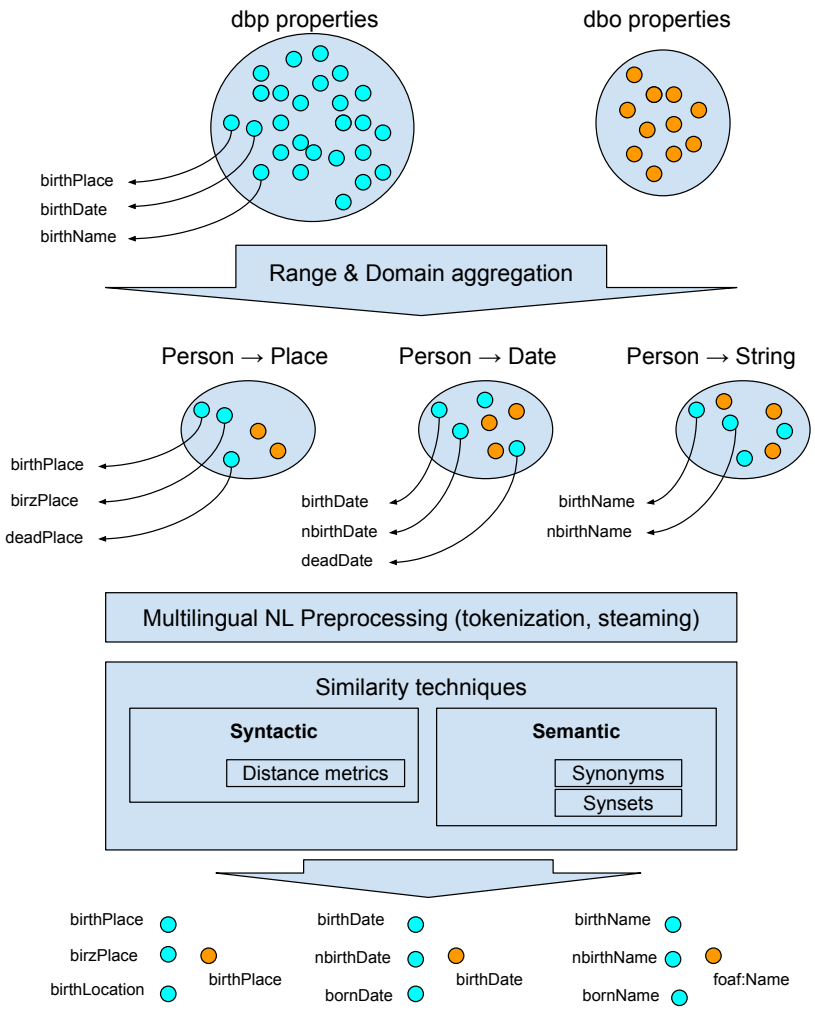

Fig. 1. Process pipeline to map dbp properties to dbo properties.

marks such as brackets (e.g. *numEmployees(globally)*) for tokenization when they were applicable. In addition, lemmatization can be used for finding more results by normalizing the different variations such the inconsistent use of singular and plural words (e.g. coachTeams → (coach, team)).

As the majority of the dbo properties only have labels in English, when non-English dbp properties are detected in DBpedia instances such as the Spanish one or the German, translation tools are used to convert the property into English for mapping with the dbo property (e.g., geburtsort → birthPlace).

The third step comprises similarity techniques. The simplest is the syntactic distance, which includes classical string distance metrics (e.g Jaro-Winkler distance, Damerau-Levenshtein distance), and token-based techniques (e.g. Jaccard similarity, Cosine Similarity). Several techniques can be used to identify different types of variations in dbp properties, for instance, edit distance-based measures such as Damerau-Levenshtein perform better for identifying typos but they are sensitive to substring locality. Using syntactic techniques such as string similarity we can identify that *dbp:birzPlace* means *dbo:birthPlace*. Semantics techniques go a step forward, and we have tested two 'semantic similarity' measures: (1) a dictionary-based method for synonyms and (2) a synsets-based method using WordNet. Semantic similarity allows us to identify that dbp properties like *dbp:birthLocation* or *dbp:cityOfBirth* are similar to the dbo property *dbo:birthPlace*. Further studies will be focused on finding the most accurate semantic-similarity methods for these tasks.

3.1 Enhancing SPARQL queries by using dbp properties

Knowing the dbp properties with the same meaning that a given dbo property, we can use them like in the example shown in listing 1.1. Here we show a simple SPARQL query containing the property *dbo:birthPlace*. Listing 1.2 shows a query enhancement based only in dbp properties syntactically similar to *dbo:birthPlace*. We use VALUES, a SPARQL 1.1 feature equivalent to a set of UNION, which allow us a more compact representation. Notice that this query uses real properties available in the English DBpedia SPARQL endpoint.

```
PREFIX dbo: <http://dbpedia.org/ontology/>
select ?s ?bp {
   ?s dbo:birthPlace ?bp .
}
```

Listing 1.1. Original SPARQL query

```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>

select ?s ?bp  where {
 ?s ?p ?bp .
 VALUES ?p {
  dbo:birthPlace #typical dbo property
  #Alternative dbp properties
  dbp:birthPlcace dbp:birthplace
  dbp:birhPlace   dbp:bithPlace
  dbp:birtPlace   dbp:biRthPlace
 }
}
```

Listing 1.2. Enhanced SPARQL query

Table 2. Example of dbp→dbo property mappings. Δ_1 is the enhancement for the example query in listing 1.1.

DBpedia	dbo prop	dbp prop		Δ_1
		Syntactic	Semantic	
English	birthPlace	birthPlace birthplace placeofbirth cityofbirth cityofbirthPlace birthPlac birthdplace birthPalce cityOfBirth birthLocation birthPlace PlaceOfBirth laceOfBirth oplaceOfBirth birthPlace. birthPlacE birthPalce birthPlae birthPace birthPlaxe birtPlace birthPlcace bithPlace brithPlace nbirthPlace birthplace birghPlace birthdplace biRthPlace birth placebirth placeOfBirth placOfBirth birthPlaceOf birthPlae		350%
Spanish	birthPlace	lugarDeNacimiento lugarNacimiento ciudaddenacimiento lugarNacimiento lugarnacimiento ciudadDenacimiento lugardenacimiento lugarNacimento paisdenacimiento paisNacimiento lugarNaciento birthPlace birthplace placeOfBirth		221%
German	birthPlace	geburtsort birthplace placeOfBirth placeofbirth	birthPlace geburtsland countryofbirth	134%

4 Evaluation example

As a complete evaluation would require more space, we only show an evaluation example to check our hypothesis that SPARQL query results can be improved by using dbp properties with the same semantics that the dbo properties used in a SPARQL query. Following the proposed method described in section 3, we use the *dbo:birthPlace* property for the analysis. Table 2 shows the possible dbp properties mapping the *dbo:birthPlace* property for the three DBpedia instances analyzed, distinguishing between syntactic and semantic techniques as described in section 3. Then, a simple query is used to analyze the number of results returned when only *dbo:birthPlace* is used (similar to listing 1.1) and when an enhanced query is used (similar to listing 1.2). This enhancement, denoted Δ_1 in the table leads to 350% improvement in the case of English DBpedia (3,940,073 results instead of 1,211,868) , 221% improvement in the case of Spanish DBpedia (765,633 results instead of 346,515), and 132% improvement in the case of German DBpedia (1,319,892 results instead of 986,323). These results illustrate that enhancing the queries using the approach proposed in this paper leads to better answers to the queries regarding the number of results. In the future, we plan to evaluate the correctness of the answers of the enhanced queries to assess if there is an impact on the quality of the results.

The queries used in the paper and the intermediate results are found in this supplementary material page[2].

[2] See `http://tinyurl.com/EKAW2016paper129extras`

5 Related work

Both Rahm and Bernstein [6], and Shvaiko and Euzenat [7] provide surveys of schema matching approaches and classify schema matching approaches into categories. The approach proposed in this paper combines several linguistic techniques that are mentioned in the survey including both syntactic and semantic techniques. Rinser et al. [8] propose a three-stage instance-based schema matching approach for mapping infoboxes from Wikipedias of different languages. The presented approach is only about Wikipedia, however it can be used to complement the property mappings proposed in this paper. Zhang et al. [9] propose Statistical Knowledge Patterns for identifying synonymous relations in large linked datasets. The method presented in this paper uses a similar technique for property clustering, but also compliment it with the NLP techniques. Aprosio et al. [10] emphasize the problem of non-mapped infoboxes in DBpedia and proposes an approach for automatic mapping generation applied to the Italian chapter of DBpedia.

6 Conclusions and future work

Our work starts by realizing that DBpedia triples are comprised not only by properties defined in the DBpedia ontology (dbo properties) but, to a big extent, by other properties (dbp properties). The DBpedia extraction process generates triples containing dbo properties when there is a mapping between a field in a Wikipedia infobox and a dbo property. But the extraction process also generates dbp properties for the fields in Wikipedia infoboxes that do not have such mapping. In the case of the English DBpedia, almost 50% of all triples contain dbp properties in its predicate. Therefore, queries containing only dbo properties cannot access big parts of the DBpedia dataset.

In order to check the infra-utilization of dbp properties, we have analyzed a SPARQL query log repository containing SPARQL queries form several datasets, concluding that our hypothesis is correct at least for the English DBpedia.

As an initial application of this work, we have sketched a method to find the most similar dbp properties for a given dbo property. This could be used to automatically enhance SPARQL queries in order to get more results and we have shown some simple usage examples.

The proposed method depends on many parameters and we have applied them to three DBpedia instances (English, Spanish and German). Future work will explore the most adequate parameters for a wider set of local DBpedia instances. For instance, we should identify the most appropriated method and parameters for syntactic similarity. A too restrictive similarity parameters would not provide much more data, and too relaxed parameters could produce wrong results. Concerning semantic similarity we have to find a similar balance. In both cases we have to test the results with real users by means of a testing tool. This tool will allow us to get the best parameters, for a given language, in order to provide the most similar dbp properties for a given dbo.

But this method is only an example of the utility of dbp properties. We claim dbp properties as first-class citizens, and linked data tools should allow users to exploit them. We show LOUPE as an exploring tool, which allows 'property exploration' for both, dbo and dbp, properties.

In summary, dbp properties are a good complement for dbo properties in SPARQL queries because they give us access to a richer DBpedia.

7 Acknowledgments

This work was funded by the JCI-2012-12719 contract, the BES-2014-068449 grant under the 4V project (TIN2013-46238-C4-2-R), JC2015-00028 and UNPM13-4E-1814.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In: The Semantic Web–ISWC 2007. Volume 4825 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2007) 722–735
2. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., et al.: DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web **6**(2) (2015) 167–195
3. Mihindukulasooriya, N., Rico, M., García-Castro, R., Gómez-Pérez, A.: An Analysis of the Quality Issues of the Properties Available in the Spanish DBpedia. In: 16th Conference of the Spanish Association for Artificial Intelligence. Volume 1., Springer International Publishing (2015) 198–209
4. Saleem, M., Ali, M.I., Hogan, A., Mehmood, Q., Ngomo, A.C.N.: Lsq: The linked sparql queries dataset. In: The Semantic Web-ISWC 2015. Springer (2015) 261–269
5. Mihindukulasooriya, N., Villalon, M.P., García-Castro, R., Gómez-Pérez, A.: Loupe-An Online Tool for Inspecting Datasets in the Linked Data Cloud. (2015)
6. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. the VLDB Journal **10**(4) (2001) 334–350
7. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. In: Journal on data semantics IV. Springer (2005) 146–171
8. Rinser, D., Lange, D., Naumann, F.: Cross-lingual entity matching and infobox alignment in wikipedia. Information Systems **38**(6) (2013) 887–907
9. Zhang, Z., Gentile, A.L., Blomqvist, E., Augenstein, I., Ciravegna, F.: Statistical knowledge patterns: Identifying synonymous relations in large linked datasets. In: International Semantic Web Conference, Springer (2013) 703–719
10. Aprosio, A.P., Giuliano, C., Lavelli, A.: Towards an automatic creation of localized versions of DBpedia. In: Proceeding of the 12th International Semantic Web Conference. Springer (2013) 494–509