Imposing a Semantic Schema for the Detection of Potential Mistakes in Knowledge Resources*

Vincenzo Maltese

DISI – University of Trento, Trento, Italy

Abstract. Nowadays, there is a pressing need for very accurate, up-to-date and diversity-aware knowledge resources. As their maintenance is very expensive, we argue that the only affordable way to address this is by complementing automatic with manual checks. This paper presents an approach, based on the notion of *semantic schema*, which aims to minimize human intervention as it allows the automatic identification of potentially faulty parts of a knowledge resource which need manual checks. Our evaluation showed promising results.

1 Defining and Enforcing a Semantic Schema

In modern ICT applications, there is a pressing need for very accurate, up-to-date and diversity-aware knowledge resources [1]. Unfortunately, so far the attempts often failed to meet the expectations [10]. While automatically built resources can scale up to millions of entities, they can hardly compete in accuracy w.r.t. manually built resources. For this reason, we argue that the only affordable way to address this is by complementing automatic with manual checks. State of the art tools focus on the automatic detection and fixing of mistakes, especially for consistency checks. Notable examples include ontology development toolkits [6] and diagnostic tools [5]. Some authors concentrate on OWL ontologies [4]; some others on RDF ontologies [7].

This paper presents an approach to the automatic detection of potential mistakes in knowledge resources, based on the notion of *semantic schema* which takes into account the meaning of the terms in the resources and enforces additional constraints on their content. As combination of machine and human processing, violations to the schema are automatically detected and directed to manual checks. We evaluated the approach on the YAGO *ontology* [2] which was selected mainly because it does not have a fixed schema, and whose 2009 version has been never evaluated before. A similar experiment, performed on the GeoNames *database*, is described in [11].

The quality of a knowledge resource can heavily depend on the strategy employed for the data representation [9]. Databases ensure certain levels of quality by enforcing integrity constraints, but they do not support the explicit encoding of domain knowledge. For instance, it is not possible to specify that what applies to generic locations also applies to lakes and mountains. Instead, the constraints that ontologies can specify depend on the expressiveness of the language used. While OWL is very expressive, RDF(S) has well-known limitations: even if it distinguishes between

* A detailed longer description of this work is available as DISI technical report with same title.

Y.T. Demey and H. Panetto (Eds.): OTM 2013 Workshops, LNCS 8186, pp. 731–734, 2013.

classes and instances, a class can be potentially treated as an instance; it is not possible to explicitly represent disjointness between classes; transitivity cannot be enforced at the level of instances. We compensate for the limitations of databases and RDF(S) ontologies by defining additional constraints that constitute what we call a *semantic schema*. The semantic schema is built on the data model described in [8] that provides the *formal language* for the schema in the following sets:

- C is a set of *classes*;
- E is a set of *entities* that instantiate the classes in C;
- R is a set of binary *relations* relating entities and classes, including *is-a* (between classes in C), *instance-of* (associating instances in E to classes in C) and *part-of* (between classes in C or between entities in E) relations. We assume *is-a* and *part-of* to be transitive and asymmetric;
- A is a set of *attributes* associating entities to the data type values.

The semantic schema is defined as a set of constraints:

- on the domain and range of the attributes in A, such that the domain is always constituted by the entities in E which are instances of one or more classes in C and the range is a standard data type (e.g. Float, String);
- on the domain and range of the relations in R, such that both the domain and range are always constituted by the entities in E which are instances of one or more classes in C;
- about known disjoint classes.

We call *types* those classes in C which are explicitly mentioned as the domain or range of a relation in R or an attribute in A. Entities in E and facts about them is what in [8] is called the *knowledge level*. It can be easily observed that the above addresses all the limitations we described for databases and RDF(S). Once the semantic schema is defined, the content of the knowledge resource is processed by enforcing the schema in two steps. With the first step each entity in the resource is assigned exactly one type X from the schema. The selection of X is performed by checking that:

1. ALL the classes associated to the entity have at least a candidate sense (a possible disambiguation) which is more specific or more general than X
2. ALL the attributes of the entity are allowed for the type X
3. X is the only type exhibiting properties 1 and 2

Entities failing this assignment are considered to violate the semantic schema and are spotted as potential mistakes. With the second step, and for those entities passing the test, corresponding classes, attributes and relations are disambiguated accordingly.

2 The YAGO Use-Case

The YAGO Ontology. YAGO is automatically built by using WordNet noun synsets and the hypernym\hyponym relations between them as backbone and by extending it with additional classes, entities and facts about them extracted from Wikipedia. The YAGO model is compliant with RDF(S). For instance, for *Elvis Presley* it includes:

Elvis_Presley	isMarriedTo	Priscilla_Presley
Elvis_Presley	bornOnDate	1935-01-08
Elvis_Presley	type	wordnet_musician_110340312
Elvis_Presley	type	wikicategory_Musicians_from_Tennessee

isMarriedTo corresponds to a relation between entities, *bornOnDate* is an attribute, and *type* connects an entity to a class, which can be a WordNet class (taken from WordNet) or a Wikipedia class (taken from Wikipedia and linked to WordNet).

Definition of the Semantic Schema. For demonstrative purposes, we focused on locations, organizations and persons. The language was defined as follows:

- C includes *location*, *person, organization*, their more specific subclasses and their more general super-classes taken from WordNet.
- E is initially empty and is later populated with entities from YAGO.
- R contains *is-a*, *instance-of*, *part-of* relations and the subset of *YAGO* relations whose domain and range intersects with the classes in C.
- A contains the subset of *YAGO* relations whose domain intersects with the classes in C and the range is a standard data type.

In order to resolve their ambiguity, the names of attributes and relations were refined, disambiguated and renamed in order to identify corresponding synsets for them in WordNet. We then defined a semantic schema where, for instance:

- **Persons, locations and organizations** can all have the following relations: {hasWebsite, hasWonPrice, hasMotto, hasPredecessor, hasSuccessor}
- **Organizations** can also have the following relations: {hasNumberOfPeople, isAffiliatedTo, hasBudget, hasRevenue, hasProduct, establishedOnDate, createdOnDate, isLeaderOf, influences, dealsWith, participatedIn, isOfGenre, musicalRole, produced, created}
- Locations, persons and organizations are pairwise disjoint.

Enforcing the Semantic Schema. Facts about locations, organizations and persons were extracted from YAGO and imported into a database. The selection of relevant knowledge was performed by using *ontology modularization* techniques [3]. Overall, we identified 1,568,080 entities that correspond to around 56% of YAGO. The classes are limited to those in C and were extracted via specifically designed NLP tools. We unambiguously assigned a type to 1,389,505 entities corresponding to around 89% of the entities extracted (case I). 20,135 entities were categorized as ambiguous because more than one type X is consistent with the classes and the attributes of the entity (case II). 158,441 entities were not categorized because of lack of information - e.g., the entity has only one class and no attributes - or conflicting information - i.e., with classes or attributes of different types (case III). Attributes and relations were mapped to attributes in A or relations in R according to the type X. Values were considered to be correct only if they were consistent with the corresponding range constraints.

Evaluation. We manually evaluated the accuracy of our class disambiguation (performed only for case I) w.r.t. YAGO on randomly selected entities. For case I, over 100 randomly selected entities our type assignment is always correct, while our disambiguation of their 250 classes is **98%** correct, which is comparable to what we found in YAGO for the same entities (**97.2%**). The mistakes tend to be the same. For case II, over 50 entities we found that the accuracy of their 65 YAGO classes is **72.3%**. Mistakes include for instance bank as river slope instead of institution.

However, **7** of those are not even entities (they include articles about events). For case III, over 50 entities we found that the accuracy of their 101 classes in YAGO is **86.14%**. Mistakes include for instance unit as unit of measurement instead of military unit. However, **72%** of the candidates present some form of wrong information or they are not even entities. They include entities which are both animals and persons; entities which are both organizations and persons; sex and political positions as locations. Thus, the evaluation confirms the need to manually inspect entities falling in cases II and III, as their quality is significantly lower than those in case I.

3 Conclusions

We presented an automatic semantic schema-based approach for the identification of those parts of a knowledge resource which are particularly noisy and therefore would benefit from manual checks. The evaluation conducted on YAGO provided promising results. The future work will include (a) an extended schema for a higher coverage on YAGO and (b) the design of crowdsourcing tasks necessary to refine noisy parts.

Acknowledgements. This research has received funding from the CUbRIK EU Project, Grant agreement no. 287704. Thanks to Fausto Giunchiglia, Biswanath Dutta, Aliaksandr Autayeu, Feroz Farazi, Mario Passamani and the other members of the KnowDive group for their help.

References

1. Giunchiglia, F., Maltese, V., Dutta, B.: Domains and context: first steps towards managing diversity in knowledge. Journal of Web Semantics 12-13, 53–63 (2012)
2. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: A Large Ontology from Wikipedia and WordNet. Journal of Web Semantics 6(3), 203–217 (2008)
3. Doran, P., Tamma, V., Iannone, L.: Ontology module extraction for ontology reuse: an ontology engineering perspective. In: 16th ACM CIKM Conference, pp. 61–70 (2007)
4. Parsia, B., Sirin, E., Kalyanpur, A.: Debugging OWL ontologies. In: WWW, pp. 633–640 (2005)
5. McGuinness, D.L., Fikes, R., Rice, J., Wilder, S.: An environment for merging and testing large ontologies. In: 7th International Conference on Principles of Knowledge Representation and Reasoning (KR 2000), pp. 483–493 (2000)
6. Noy, N., Sintek, M., Decker, S., Crubezy, M., Fergerson, R., Musen, M.: Creating semantic web contents with Protégé-2000. In: IEEE Intelligent Systems (2000)
7. Ding, L., Finin, T.W.: Characterizing the Semantic Web on the Web. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 242–257. Springer, Heidelberg (2006)
8. Giunchiglia, F., Dutta, B., Maltese, V., Farazi, F.: A facet-based methodology for the construction of a large-scale geospatial ontology. Journal of Data Semantics 1(1), 57–73 (2012)
9. Martinez-Cruz, C., Blanco, I., Vila, M.: Ontologies versus relational databases: are they so different? A comparison. Artificial Intelligence Review, 1–20 (2011)
10. Jain, P., Hitzler, P., Yeh, P.Z., Verma, K., Sheth, A.P.: Linked data is merely more data. In: Linked Data Meets Artificial Intelligence, pp. 82–86 (2010)
11. Maltese, V., Farazi, F.: A semantic schema for GeoNames. In: INSPIRE Conference (2013)