Enforcing a semantic schema to assess and improve the quality of knowledge resources [1]

Vincenzo Maltese

DISI – University of Trento, Trento, Italy
Phone: +39.0461.283824 - maltese@disi.unitn.it

Abstract. Modern Information and Communication Technologies (ICT) require very accurate and up-to-date knowledge resources, such as databases and knowledge bases, providing information about real world entities (e.g. locations, persons, events) which can guarantee that results of automatic processing can be trusted enough for decision making processes. The solutions employed so far to guarantee for their quality mainly rely on the automatic application of integrity constraints for databases and consistency checks for knowledge bases. In order to achieve a higher accuracy, there is also a recent trend in complementing automatic with manual checks, via crowdsourcing techniques. This paper presents a methodology and an evaluation framework, based on the definition and application of a *semantic schema*, which analyses the (sometimes hidden) semantics of the terms in the entity descriptions from a knowledge resource, and allows assessing its quality and the identification of those potentially faulty parts which would benefit from manual checks. The approach is particularly suited for schema-less resources, i.e. resources in which entities do not follow a unique and explicit schema. Our evaluation showed promising results.

Keywords: knowledge resources, data semantics, data quality, knowledge evaluation, ontologies, semantic schema

1 Introduction

In modern society there is a pressing need for very accurate and up-to-date knowledge resources, namely any dataset providing knowledge about entities of the real world (e.g. locations, persons, events), ranging from databases to knowledge bases. Such knowledge is fundamental for a better understanding of the world both at individual (e.g. by citizens and tourists for daily planning) and societal levels (e.g. by policy makers for city planning). The Open Government Data[2] initiative, aiming at making public and distributing the content of institutional datasets, is a clear example of such a need. For instance, by providing information about local transportation means and

[1] This paper extends with significantly more details the work presented by the same author at the ODBASE 2013 conference [21].

[2] http://opengovernmentdata.org/

touristic attractions of a city, a knowledge resource can be employed by automatic tools to provide real-time information about optimized (e.g., in terms of personal interests, as well as price and time required) sightseeing routes to tourists and suggestions for better (e.g., in terms of reduced traffic and pollution) transportation planning to policy makers.

Unfortunately, so far the attempts to achieve positive outcomes from such systems often failed to meet the expectations [17]. Above all, the Semantic Web [18] has been only marginally successful because not being yet capable of providing usable resources, namely something up-to-date and reliable enough for decision making processes. In fact, while automatically built resources (e.g., developed by extracting knowledge from the Web) can scale up to millions of entities, they can hardly reach the accuracy of manually built resources. Conversely, the latter are typically pretty small and expensive to keep updated. In general, there is also a fragmentation of such resources which turn out to be hardly reusable and interoperable [22, 17].

This paper presents a methodology and an evaluation framework, based on the definition and application of a *semantic schema* (in opposition to database schemas which are syntactic as there is no explicit meaning attached to them) which analyses the (often hidden) semantics of the terms occurring in the entity descriptions from a knowledge resource, and allows identifying those potentially faulty descriptions which would benefit from further manual checks. The semantic schema looks at the overall consistency of *all* the assertions associated to a certain entity (rather than looking just at the consistency of individual assertions) and gives meaning to the terms (classes, relations and attribute names) accordingly, and whenever possible. It also ensures that entities have an unambiguous semantics via the usage of explicit disjointness constraints between classes, and constraints on the domain and range of the attributes. Violations to the schema are automatically detected. The approach is particularly suited for schema-less resources, i.e. resources in which entities do not follow a unique and explicit schema. We evaluated the approach on the YAGO *ontology* [3] which was expressively selected because it does not have a fixed schema, and because its 2009 version was been never evaluated before. A similar experiment has been performed with GeoNames [19]. For example, by enforcing the semantic schema on the following entity description taken from YAGO:

Bank of Belize	type	bank
Bank of Belize	establishedOnDate	1982-01-05

it can determine that the only meaning of the term bank which can guarantee the overall consistency of this entity description is *bank as institution* and not *bank as sloping land* as it is wrongly associated in YAGO (a sloping land should not have a date of establishment). By enforcing the schema on the following entity description:

Alvin McDonald	type	hill
Alvin McDonald	type	person
Alvin McDonald	diedOnDate	1893-12-15

it can determine that it should contain some mistakes since locations and persons are disjoint.

Notice that YAGO already disambiguates the classes and imposes domain-range constraints at the level of single facts, but no attention is paid to their overall coherence when the facts aggregate into a description of a single entity. This is where a semantic schema can make the difference. Another way to understand this work it to consider the semantic schema as a way to "semantify" an entity description, namely to make explicit the hidden semantics of terms thus identifying the meaning that better suits each single term given the overall coherence of all facts and terms used across facts.

The rest of the paper is organized as follows. Section 2 provides some relevant state of the art. Section 3 introduces the notion of semantic schema and explains how to enforce it via the evaluation framework for the detection of potential mistakes. Section 4 briefly describes the YAGO ontology. Section 5 focuses on the definition of a semantic schema for YAGO. Section 6 explains how the dataset was prepared in order to enforce the schema as it is described in Section 7. Section 8 provides the evaluation. Finally, Section 9 concludes the paper by summarizing the work done and outlining the future work.

2 State of the art

Large-scale knowledge resources. In the recent years, several knowledge resources have been built, manually or automatically. WordNet[3], Cyc[4] and SUMO[5] are examples of manually built resources. WordNet is focused on linguistic information and it is by far the most widespread. Though, among its drawbacks we can mention that it is not tailored for any particular domain, it is often considered too fine grained to be really useful [14], and it contains a very small number of entities [20]. In fact, resources of this kind tend to be accurate, but quite small in size. Domain specific resources are offered by Library Science communities, but they typically lack of explicit semantics [15]. Among (semi-)automatically generated resources we can mention DBPedia[6], YAGO and Freebase[7] offering millions of entities and facts about them. Resources of this kind tend to be much bigger in size. Assessing their quality is clearly fundamental in critical domains such as transportation and health.

Tools for the evaluation and improvement of knowledge resources. The quality of the data contained in knowledge resources can heavily depend on the strategy employed for the data representation [16]. Databases ensure certain levels of data quality by enforcing integrity constraints, but it is not possible to directly codify domain knowledge in them (e.g. in terms of formal ontologies). For instance, it is not possible to apply a constraint to a class and all its more specific classes, e.g. the fact that what is enforced for generic locations is also enforced for lakes and mountains. In effect, they do not take into account the meaning of the terms. On the other hand, the

[3] http://wordnet.princeton.edu/

[4] http://www.cyc.com/

[5] http://www.ontologyportal.org/

[6] http://dbpedia.org/

[7] http://www.freebase.com/

constraints that can be specified in knowledge bases depend on the expressiveness of the language used. While the OWL language is extremely powerful, the RDFS model has well-known limitations: even if it distinguishes between classes and instances, a class can be potentially treated as an instance [8]; it is not possible to explicitly represent disjointness between classes; transitivity cannot be enforced at the level of instances [2].

Several tools have been developed at the purpose of evaluating knowledge bases. Syntax and consistency checks are typically performed by ontology development toolkits, such as Protégé [12]. Preece and Shinghal [9] provide a survey of the programs used to anomaly detection, where an anomaly is the sign of probable errors. Several diagnostic tools have been developed. For instance, ODEval [4] detects potential taxonomical errors in terms of *inconsistency* (when contradictory conclusions can be obtained), *incompleteness* (when it does not fully capture what it is supposed to represent of the real world) and *conciseness* (when it contains redundancies). Ceuster et al. [11] present a semi-automatic approach to the detection and repair of anomalies in medical ontologies. It mainly focuses on incompleteness in terms of potentially missing relations, classes and entities. The Chimaera suite [10] offers checks for incompleteness, taxonomic analysis, and semantic evaluation. The effectiveness of such tools depends on the expressiveness of the language used to represent the source. Differently from our approach, all these tools assume (a) that the schema has been already defined and (b) that the meaning has been already assigned to the terms occurring in the source and therefore they are not directly usable in contexts with hidden semantics, i.e. where it is also necessary to determine, verify or improve the quality of the disambiguation of the terms denoting classes, relations, attributes and their values (as we do).

3 Defining and enforcing the semantic schema

We compensate for the limitations of existing knowledge resources by defining a *semantic schema*. Higher level quality control software modules are implemented to guarantee that the facts associated to the same entity are collectively consistent w.r.t. the additional constraints, even though these constraints cannot be expressed by the representation language originally employed (i.e. in the case of databases and RDFS).

Our approach is based on an *evaluation framework* which assists the user during the various phases of the process. The framework mainly consists in a *collaborative platform* we developed that allows the definition of a set of entity types (the semantic schema) and the corresponding terminology that can be defined by several experts in a collaborative way [24], and a *relational database* that hosts the data and where the semantic schema is enforced and checked. The phases of the process are described below.

3.1 Identification of the knowledge resource

One or more knowledge resources which are judged as useful to support some specific reasoning tasks are identified. This can be done on the basis of several criteria such as coverage in terms of specific types of entities or specific attributes which are required. For instance, in case the reasoning task consists in the discovery of the facilities of a given city, then Open Data from the local government integrated with Open Street Map[8] can serve the purpose as they may provide the list of facilities of various kinds as well as their latitude and longitude coordinates.

3.2 Definition of the semantic schema

Using the collaborative platform, we build a semantic schema which follows the data model initially described in [13] and further refined in [25] that provides the corresponding *natural* and *formal language* which are required to define it. In the *formal language*, we identify 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 the canonical *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 data type values.

Each element in the formal language is associated to a set of words and a gloss from a *natural language* vocabulary. The evaluation framework was initially populated with the English language taken from WordNet 2.1, but it can be extended at any time as needed by using the collaborative platform.

We then define a semantic schema as a set of *entity types* $T = \{T_1,\dots,T_n\}$, where each entity type T_i is assigned a class c_i and its more specific classes from C, a subset of the attributes in A and a subset of the relations in R, as well as a corresponding set of constraints:

- on the domain and range of the attributes, such that the domain is always constituted by the entities in E which are instances of c_i (or more specific class) and the range is a standard data type (e.g. Float, String);
- on the domain and range of the relations in R, such that the domain is always constituted by the entities in E which are instances of c_i (or more specific class) and the range is constituted by the entities in E which are instances of a class c_j of a corresponding entity type $T_j \in T$;
- about the entity types which are mutually disjoint ($T_i \perp T_j$).

Entities in E and facts about them constitute what in [13] is called the *knowledge*. It can be easily observed that the above addresses all the limitations we described for databases and RDFS. With respect to the former, the formal language provides the meaning of the terms taken from the natural language and constitutes the domain knowledge, while the semantic schema provides constraints that apply to a class and all its more specific classes. With respect to the latter, the semantic schema enforces a

[8] https://www.openstreetmap.org

clear split between entities (E) and classes (C), thus preventing a class to be used as source or target of a relation; it enforces disjointness between classes and transitivity between entities when it is the case.

Though a default one might be used, adapted or extended, the semantic schema is defined on purpose according to the expected content of the knowledge resource and reflects the mental model of the modeler. Therefore, the final result should be always considered as relative to a particular vision of the world. This may require an initial inspection of the content of the knowledge resource in terms of kinds of entities, their classes, attributes and relations. For instance, if it is supposed to contain knowledge about locations and persons we might define the language where C contains *location*, *person* and their more specific subclasses (e.g., *city*, *river* and *hill* for location; *professor*, *student* and *scientist* for person); E contains actual locations and persons (e.g., *Rome* the city and *Albert Einstein* the scientist); R contains *is-a*, *instance-of*, *part-of* and *birthplace* relations; A contains *latitude*, *longitude* and *birthdate* attributes. We might then define the semantic schema T = {location, person} where:

- **locations** can have *latitude* and *longitude* which are Floats, *part-of* relations between them, and nothing else;
- **persons** can have a *birthdate* which is a Date, a *birthplace* which is a *location*, and nothing else;
- locations and persons are disjoint.

3.3 Importing the knowledge resource in a relational database

In order to be able to enforce the semantic schema, relevant facts are extracted and preliminary imported into a relational database. As the database does not yet comply with the semantic schema, we call this database the *intermediate schema*. In particular, facts are represented as classes, attributes and relations which are not yet disambiguated, and the attribute values are not yet checked for consistency w.r.t. the constraints provided by the schema. This step typically requires the development of dedicated extractors that depend on the structure and the language originally used to codify the knowledge resource. The database has the following generic schema:

Entity (entityID, etypeID, name)
Class (classID, name)
EntityClass (entityID, classID, classnameID)
Property (entityID, PropertyID, name, value)

Entity stores the entities and its name extracted from the original knowledge resource; name denotes the preferred name of the entity, while additional names can be stored as values of an attribute; etypeID is initially empty and it is set once the entity type is discovered (it corresponds to one of the entity types in the schema or can be left as undefined if it cannot be found). *Class* stores information about the class names. *EntityClass* associates each entity to its classes; classnameID is initially empty and it is set once the class name is disambiguated independently for each single entity (it corresponds to one of the classes in C). Property stores names and values of the attributes and relations; PropertyID is initially empty and it is set once the attribute/relation name is disambiguated (it corresponds to one of the attributes in A or one of the relations in R).

3.4 Enforcing the semantic schema on the knowledge resource

Once the relational database has been filled with data coming from the original knowledge resource, the database is processed by enforcing the semantic schema in two steps. With the first step each entity in the resource is assigned exactly one entity type X from the schema. This is done by setting etypeID = X for the corresponding entity in the database. The selection of X is performed by an algorithm that simply checks that all the following conditions are met w.r.t. the schema and the formal language defined for it:

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 (the class of) X
2. ALL the attributes and relations of the entity are allowed for X
3. X is the only entity type exhibiting properties 1 and 2

Entities failing this test (i.e. they do not fit in any X or they fit in more than one X) are considered to violate the semantic schema and are spotted as potential mistakes. With the second step, and only for those entities passing the test, meaning is given to the terms in the entity descriptions: thus classes, relations and attribute names are disambiguated accordingly. This means that we use classes, relations and attributes as contextual information to restrict candidate senses (in other words, they disambiguate each other). Specifically, we always assign to classes the WordNet sense more specific (or more general) than the class of the entity type X. In case more than one with such property is available, we assign the sense with highest rank among them. This is done by setting classnameID with the corresponding class in C in the database. Similarly, attributes and relations are mapped to the corresponding attribute in A or relation in R according to the entity type X assigned. This is done by setting propertyID with the corresponding attribute in A or relation in R in the database. Values are considered to be correct only if they are consistent with the corresponding range constraints.

For instance, with the first step an entity with classes *printer* and attribute *bornOnDate* is associated to *person* (despite printer may also mean a device). With the second step printer is disambiguated as *"someone whose occupation is printing"* and bornOnDate as *"the date on which a person was born"*.

Notice that the tests above can only fail (1) in case there is not enough natural and formal language providing possible meanings of a certain class name or (2) the natural language is too fine-grained and too many senses are provided for the same term. As example of the first, in WordNet 2.1 there is only one sense for the term *derby* as more specific than artifact defined as *"a felt hat that is round and hard with a narrow brim"*, and therefore the algorithm above can never recognize it as sporting event. As example of the second, in WordNet 2.1 there is a sense of dog as more specific than person defined as *"a dull unattractive unpleasant girl or woman"* that may determine that some animals might be disambiguated as people by mistake. In fact, failures indicate ways by which the formal and natural language can be improved.

3.5 Evaluation of the results

The final step consists in selecting a random subset of the entities from the database and in verifying that entity types, classes, relations and attributes are assigned and disambiguated correctly when it does not fail and that the original knowledge resource contained mistakes when it fails. The high quality data can be directly employed for the envisioned reasoning tasks. The low quality data is manually inspected to fix the mistakes or simply ignored.

4 The YAGO ontology

The YAGO ontology [3] 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 infoboxes and categories. The YAGO model is compliant with RDFS. Entities are therefore described in terms of facts of the kind <source, relation, target>. Overall, 95 different relation kinds are instantiated in YAGO 2009 version generating around 29 million facts about 2.5 million entities. Quality control is guaranteed by ensuring that facts are individually consistent with the domain and range defined for the relations. For instance, for the entity *Elvis Presley* YAGO includes the following facts:

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

where *isMarriedTo* corresponds to a relation between entities, *bornOnDate* is a data attribute, and *type* connects an entity to a class. Classes are of three different kinds:

- ***WordNet classes***, with prefix “wordnet_”, correspond to WordNet synsets;
- ***Wikipedia classes***, denoted with the prefix “wikicategory_”, correspond to Wikipedia categories which are linked to WordNet classes;
- ***YAGO classes***, such as “YagoInteger”, are additional classes introduced to enforce type checking on the domain and range of the relations.

The linking of Wikipedia with WordNet classes is automatically computed by extracting and disambiguating the head of the sentence from Wikipedia categories. In most of the cases, as senses in WordNet are ranked, the first sense is assigned to all the occurrences of the same word across the whole ontology. For instance, the head of the category *Musicians from Tennessee* is *musician* that is disambiguated as wordnet_musician_110340312. The same meaning is assigned to all the occurrences of musician in the entire ontology. The linking between Wikipedia and WordNet classes is maintained via the *subClassOf* relation.

With respect to YAGO, our approach differs in two main aspects: by employing a semantic schema (1) the consistency of facts is checked at the level of the whole entity, rather than just one fact at a time; (1) the disambiguation of the terms (classes, relations, attribute names) is performed locally to an entity description by taking into account their overall consistence that determines the entity type assigned to the entity, rather than assigning a unique disambiguation to all the occurrences of the same term across the whole dataset.

5 Definition of a semantic schema for YAGO

As YAGO is a large-scale ontology not targeted to any specific domain, it is pretty hard to define a semantic schema to capture the whole content of YAGO. Therefore, for demonstrative purposes, we decided to define a schema covering only a portion of YAGO. This proves the applicability of the approach, still requiring extending the schema in the future. We decided to focus on *locations*, *organizations* and *persons*. We can assume for instance that these are the only entity types that are required for a specific reasoning task we need to perform. To come up with a meaningful schema, we inspected its content and analyzed the definition of the relation kinds as they are given in the YAGO documentation. In doing so, we faced the following issues:

- **Lack of explicit semantics:** no explicit meaning of the attribute and relation names is given thus we found them to be ambiguous. For instance, the YAGO relation *hasHeight* can be interpreted as *height* (in the sense of stature) in case of persons, *altitude* in case of locations and *tallness* for buildings. They correspond to three different senses in WordNet. In fact, they are three different attributes. We address this problem by assigning a different sense from WordNet to each of the attributes and relations in our schema. Wikipedia classes also lack of explicit definition, thus they are ambiguous too. For instance, consider the class *Cemeteries in Canada*. There are several locations in the world named Canada, not necessarily the country. For this reason, we decided to only focus on WordNet classes.
- **Too broad domain and range**: in some cases, the domain and range of the defined relation kinds look too broad. For instance, the relation *isAffiliatedTo* is defined between any two generic entities; we rather believe that the domain should include only persons and organizations while the range should only include organizations. We address this by refining them in our schema.
- **Lack of latitude and longitude coordinates**: locations lack of latitude and longitude. As they would be extremely useful to determine whether an entity is a location, we extracted them from Wikipedia. Among available ones, we took the closest Wikipedia dump to the one used by the YAGO version used. The heuristics we used allowed us extracting 440,687 latitude\longitude pairs.
- **Lower than expected accuracy of the linking of Wikipedia to WordNet classes.** By analyzing 500 Wikipedia classes randomly taken from YAGO we found out that the accuracy of the linking is 82% (in the worst case) which is lower than the 95% claimed on average for the other YAGO versions.

Additional details are given in the appendix. We take YAGO without meaning attached to terms as their disambiguation is integral part of the application of our approach.

After the initial inspection, we defined a ***formal language*** as follows:

- C contains *location*, *person*, *organization*, their more specific subclasses and their more general super-classes (e.g. *entity*, *physical object*) from WordNet. As we assume *facilities*, *buildings, bodies of water*, *geological formations*, *dry lands* and *geo-political entities* (such as countries and cities) to be locations, corresponding more specific subclasses are also contained in C. The set C does not contain Wikipedia classes.
- E is initially empty and it is later populated with entities from YAGO.
- R contains *is-a*, *instance-of*, *part-of* (the YAGO *locatedIn*) relations and the subset of *YAGO* relations whose domain and range intersects with the classes in C and where such relations were refined, disambiguated and renamed in order to identify corresponding synsets for them in WordNet. For instance, the YAGO relation *isAffiliatedTo* was renamed as *affiliation* defined in WordNet as "a social or business relationship", the domain was restricted to the union of person and organization and the range to organization.
- A contains the subset of *YAGO* relations whose domain intersects with the classes in C, the range is a standard data type and where such relations were refined, disambiguated and renamed in order to identify corresponding synsets for them in WordNet. For instance, *hasHeight* (being ambiguous) is mapped to the attribute *height* (in the sense of stature) in case of persons, *altitude* in case of generic locations and *tallness* for buildings. A is extended with *latitude* and *longitude* whose domain is location and range is Float.

The ***natural language*** corresponds to all the terms and synsets in English taken from WordNet 2.1 which correspond one by one to the classes, relations and attributes in the formal language.

We then defined the ***semantic schema*** T = {location, person, organization, geo-political entity, facility, building} where:

- **Persons, locations** and **organizations** can all have the attributes\relations corresponding to the following YAGO relations: {hasWebsite, hasWonPrize, hasMotto, hasPredecessor, hasSuccessor}
- **Persons** can also have: {hasHeight, hasWeight, bornOnDate, diedOnDate, bornIn, diedIn, originatesFrom, graduatedFrom, isAffiliatedTo, isCitizenOf, worksAt, livesIn, hasChild, isMarriedTo, isLeaderOf, interestedIn, influences, isNumber, hasAcademicAdvisor, actedIn, produced, created, directed, wrote, discovered, madeCoverOf, musicalRole, participatedIn, isAffiliatedTo, politicianOf}
- **Organizations** can also have: {hasRevenue, hasBudget, dealsWith, produced, created, hasNumberOfPeople, isAffiliatedTo, musicalRole establishedOnDate, hasProduct, isLeaderOf, createdOnDate, influences, participatedIn, isOfGenre}
- **Locations** can also have the following: {latitude, longitude, hasHeight, hasUTCOffset, establishedOnDate, hasArea, locatedIn, inTimeZone}

- **Geo-political entities** are more specific locations that can also have: {hasGini, hasPoverty, hasCapital, imports, exports, hasGDPPPP, hasTLD, hasHDI, hasNominalGDP, hasUnemployment, isLeaderOf, has_labour, dealsWith, has_imports, has_exports, has_expenses, hasPopulation, hasPopulationDensity, participatedIn, hasCurrency, hasOfficialLanguage, hasCallingCode, hasWaterPart, hasInflation, hasEconomicGrowth}
- **Facilities** and **buildings** are more specific locations that can also have: {hasNumberOfPeople, createdOnDate}
- Locations, persons and organizations are pairwise disjoint.

6 Importing YAGO into the relational database

In order to be able to enforce the semantic schema, relevant facts about locations, organizations and persons taken from YAGO were extracted and preliminary imported into the relational database. The selection of relevant knowledge was performed by following principles at the basis of *ontology modularization* techniques. d'Aquin et al. [6] define ontology modularization as the task of partitioning a large ontology into smaller parts each of them covering a particular sub-vocabulary. Doran et al. [7], define an ontology module as a self-contained subset of the parent ontology where all concepts in the module are defined in terms of other concepts in the module, and do not refer to any concept outside the module. They reduce module extraction to the traversal of a graph given a starting vertex that ensures in particular that the module is transitively closed w.r.t. the traversed relations. Cuerca Grau et al. [23] stress that the partitioning should preserve the semantics of the terms used, i.e. the inferences that can be made with the terms within the partition must be the same as if the whole ontology had been used. We understand modularization as the process of identifying self-contained portions of the ontology providing the terminology which is needed to define certain specific entity types. In our work locations, organizations and persons were taken from YAGO by selecting all those entities whose WordNet class (identified through the *type* relation) is equivalent or more specific (identified through the *subClassOf* relation) than one of those in Table 1. The table also shows the amount of entities and Wikipedia classes found in each sub-tree.

WordNet Class	Entities	Wikipedia classes
wordnet_location_100027167	412,839	16,968
wordnet_person_100007846	771,852	67,419
wordnet_organization_108008335	213,952	19,851
wordnet_facility_103315023	83,184	8,790
wordnet_building_102913152	49,409	6,892
wordnet_body_of_water_109225146	36,347	1,820
wordnet_geological_formation_109287968	19,650	1,978
wordnet_land_109334396	8,854	805

Table 1. Number of entities and Wikipedia classes for the WordNet classes

Overall, we identified 1,568,081 unique entities that correspond to around 56% of YAGO. Selected entities, corresponding classes, alternative names in English and Italian and other related facts codifying their attributes and relations were then imported into the *intermediate schema*. Table 2 provides corresponding statistics. Notice that entities can belong to more than one class and this explains why the mere sum is bigger than 1,568,081.

Kind of object	Amount
Classes	3,966
Entities	1,568,080
instance-of relations	3,453,952
Attributes/Relations	3,229,320
Alternative English names	3,609,373
Alternative Italian names	220,151

Table 2. Kind and amount of objects in the intermediate schema

Notice that the classes in Table 2 do not correspond to the original Wikipedia classes as we recomputed them. In doing so, we directly associated entities to classes likely to correspond to those in C because syntactically matching with words in WordNet synsets. The class extraction was performed through the use of NLP tools, and specifically of a POS tagger developed and trained with the work presented in [5], and a BNF grammar generated to work on POS tagged Wikipedia classes. From our experiments the grammar turns out to be able to process from 96.1 to 98.7% of them according to the different sub-tree in which they are rooted, where the roots are the WordNet classes listed in Table 2. For the uncovered cases, we reused the YAGO linking. The final grammar, able to recognize *class names* and *entity names* appearing in Wikipedia classes, is as follows:

```
wikipedia-class ::= classes IN [DT] [pre-ctx] entity {post-cxt}* | classes
classes ::= class [, class] [CC class]
class ::= {modifier}* class-name
class-name ::= {NNS}+ | NNS IN {JJ}* NN [^NNP]
modifier ::= JJ | NN | NNP | CD | VBN
entity ::= {NNP}+ | CD {NNP}*
pre-ctx ::= ctxclass IN
post-ctx::= VBN IN {CD | DT | JJ | NNS | NN | NNP}* | CD | , entity | ctxclass |
             (ctxclass) | (entity [ctxclass])
ctxclass ::= {NN}+
```

For instance, from the Wikipedia class *City, towns and villages in Ca Mau Province* the grammar allows extracting the three classes *city*, *town* and *village* (while YAGO extracts *city* only), while from *Low-power FM radio stations* the grammar allows extracting *radio station* (while YAGO extracts *station* only). When multiple classes are extracted from a Wikipedia class, modifiers of the first class are assumed to apply to all classes. For instance, *Ski areas and resorts in Kyrgyzstan* means *Ski areas and ski resorts in Kyrgyzstan*. Some modifiers can explicitly (with NNP) or implicitly (with JJ) denote a named entity and are therefore filtered out. An example for the first kind is *Hawaii countries*, while an example of the second kind is *Russian subdivisions*. Less frequent POS tags found (e.g. NNPS and VBG) were not included in the grammar.

7 Enforcing the semantic schema on YAGO

As reported in Table 3, by enforcing the defined schema we could unambiguously assign an entity type to 1,389,505 entities corresponding to around 89% of the entities in the intermediate schema (case I); 20,135 entities were categorized as ambiguous because more than one entity 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 or conflicting information (case III). For those entities passing the test, entity classes, relations and attributes were disambiguated and assigned elements in C, R, and A respectively according to the entity type X associated to them.

Type	**Amount**
Person	719,551
Organization	154,153
Location	284,267
Geological formation	14,426
Body of water	34,958
Geo-political entity	100,910
Building and facilities	81,240
Total:	**1,389,505**

Table 3. Type assignment to the entities in the intermediate schema

8 Evaluation

With the initial selection 1,568,081 entities and related facts were extracted from YAGO and imported into the intermediate schema. By enforcing the semantic schema:

- **CASE I:** 1,389,505 entities (around 89% of the imported entities) were assigned exactly one entity type X;
- **CASE II:** 20,135 entities were assigned as ambiguous, i.e. more than one entity type is consistent with the classes and the attributes of the entity;
- **CASE III:** 158,441 entities were not categorized because of lack of or conflicting information.

We then evaluated the quality of our class disambiguation w.r.t. the one in YAGO 2009. Notice that as we recomputed the entity classes by extracting them from the Wikipedia classes, they might differ, also in number, w.r.t. those in YAGO. Notice that classes were disambiguated only for those entities passing the test (case I). For them, the accuracy of the entity type assignment was also evaluated.

CASE I. Over 100 randomly selected entities our assignment of the entity type turns out to be always correct, while our disambiguation of their 250[9] classes is **98%** correct (5 mistakes). By checking the Wikipedia classes of these entities in YAGO, we found out that the corresponding linking of their 216 classes is **97.2%** correct (6 mistakes). The mistakes tend to be the same, for instance we both map crater to volcanic crater instead of impact crater; manager to manager as director instead of manager as coach; captain to captain as military officer instead of captain as group leader. An example of (correct) entity description falling is this case is the following:

William Thomas Calman	type	employee
William Thomas Calman	type	zoologist
William Thomas Calman	bornOnDate	1871-12-29
William Thomas Calman	diedOnDate	1952-09-29

CASE II. 50 random entities were selected among those that we categorized as ambiguous. We found out that the accuracy of the linking of their Wikipedia classes in YAGO is **72.3%** (18 mistakes over 65). Mistakes include for instance bank as river slope instead of institution; ward as person instead of administrative division; carrier as person instead of warship; division as military division instead of geographical. Yet, **10%** of these entities (5 over 50) are neither locations, nor organizations, nor persons. They are in fact reports about the participation of some country to some competition while they are treated as countries (e.g., *Norway in the Eurovision Song Contest 2005*). Further **4%** of them (2 over 50) are not even entities. One is actually a list of categories and the other a list of countries. Overall **14%** of them look therefore wrong. In addition, among the 72.3% of the cases considered as correct there are actually 18 controversial cases where it is not clear if the class is really meant as geographical or political division (e.g. subdivision); it is not clear if the class is meant as geographical or political entity (e.g. country); it is not clear if the class is meant as organization or building (e.g. hospital). We believe that these cases might be due to the phenomenon of *metonymy* which generates very fine-grained senses in WordNet. Solving this problem would result in better disambiguation. An example of entity description falling is this case (because globally consistent both as location and organization) is the following:

Russia in the Eurovision Song Contest	type	country
Russia in the Eurovision Song Contest	hasWebsite	http://evrovid.rutv.ru

[9] Notice that the accuracy of other YAGO versions was evaluated on a much smaller sample. As it can be clearly seen from the statistics section of the YAGO website (http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/), each relation kind is typically evaluated via less than 100 instances (only 55 for the YAGO *type*).

CASE III. 50 random entities were selected among those that we preferred to do not assign any type because of lack of information (e.g. the entity has only one class and no attributes) or presence of conflicting information (i.e. classes or attributes of different types). We found out that the accuracy of the linking in YAGO for these cases is **86.14%** (14 mistakes over 101). Mistakes include unit as unit of measurement instead of military unit; model as fashion model instead of mathematical model. Though, **72%** of the candidates (36 over 50) contain mistakes or they are not even entities. They include for instance entities which are both animals and persons (e.g., we found 137 persons as fishes and 4,216 as dogs); entities which are both organizations and persons; or even sex and political positions marked as locations. An example of entity description falling is this case (because of presence of conflicting information as the attribute indicates that it should be a person, while the only meaning available in WordNet for argentine as noun is *"any of various small silver-scaled salmon-like marine fishes"*) is the following:

Ernestina Herrera de Noble	type	argentine
Ernestina Herrera de Noble	bornOnDate	1925-06-07

Thus, the evaluation confirms the need to manually inspect entities falling in case II and case III as their quality is significantly lower than those in case I.

9 Conclusions

Starting from the observation that individual and societal decision making processes require very accurate, up-to-date and knowledge resources, we presented an automatic semantic schema-based approach to "semantify" and assess their quality and for the identification of those parts of a knowledge resource which are particularly noisy and need, with higher priority w.r.t. other parts, to be manually inspected and fixed. Dually, it allows identifying those parts of higher quality that can be already trusted enough. In this way, human involvement (very costly in general) can be reduced by directing the attention to the lower quality parts.

The approach was evaluated on the YAGO ontology in its 2009 version, a large-scale knowledge base not targeted to any specific domain, with no fixed schema and never evaluated previously. As proved by the final figures, the definition and enforcement of the semantic schema allowed identifying those portions of the ontology (knowledge falling in case II and III) which are particularly noisy and that would benefit from further (manual) refinement. Higher quality portions (knowledge falling in case I) have been selected and imported into Entitypedia [1], a knowledge base under development at the University of Trento.

Future work can focus on the extension of the semantic schema to have a higher coverage on YAGO (e.g., by defining types for books, movies, events) and the design of crowdsourcing tasks necessary to refine the potentially noisy parts. More in general, we plan to develop some standard entity type semantic schemas to be used as starting point in specific domains and to extend the evaluation framework to be used in a broader range of application scenarios.

Acknowledgements. The research leading to these results has received funding from the CUbRIK Collaborative Project, partially funded by the European Commission's 7th Framework ICT Programme for Research and Technological Development under the Grant agreement no. 287704. Thanks to Fausto Giunchiglia for his suggestions, Biswanath Dutta for the early work in analyzing YAGO, Aliaksandr Autayeu for his support for the NLP tools, Feroz Farazi, Mario Passamani and the other members of the KnowDive group for the technical assistance.

References

1. Giunchiglia, F., Maltese, V., Dutta, B. (2012). Domains and context: first steps towards managing diversity in knowledge. Journal of Web Semantics, 12-13, 53-63.
2. Maltese, V., Farazi, F. (2011). Towards the Integration of Knowledge Organization Systems with the Linked Data Cloud. UDC seminar.
3. Suchanek, F. M., Kasneci, G., Weikum, G. (2008). YAGO: A Large Ontology from Wikipedia and WordNet. Journal of Web Semantics, 6 (3), 203-217.
4. Corcho, O., Gomez-Perez, A., Gonzalez-Cabero, R., and Suarez-Figueroa, C. (2004). ODEVAL: a tool for evaluating RDF(S), DAML + OIL, and OWL Concept Taxonomies. In Proceedings of the 1st Conference on AI Applications and Innovations, 369–382.
5. Autayeu, A. Giunchiglia, F., Andrews, P. (2010). Lightweight parsing of classifications into lightweight ontologies. In Proceedings of the European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2010).
6. d’Aquin, M., Schlicht, A., Stuckenschmidt, H., Sabou, M. (2007). Ontology modularization for knowledge selection: experiments and evaluations. In Proceedings of the 18th International Conference on Database and Expert Systems Applications, 874-883.
7. Doran, P., Tamma, V., Iannone, L. (2007). Ontology module extraction for ontology reuse: an ontology engineering perspective. In Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM), 61-70.
8. Brickley, D., Guha, R. V. (2004). RDF Vocabulary Description Language 1.0: RDF Schema, (Editors), W3C Recommendation.
9. Preece, A. D., Shinghal, R. (1994). Foundation and application of knowledge base verification. International Journal of Intelligent Systems, 9 (8), 683–701.
10. McGuinness, D.L., Fikes, R., Rice, J., Wilder, S. (2000). An environment for merging and testing large ontologies. In Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000), 483–493.
11. Ceusters, W., Smith, B., Kumar, A., Dhaen, C. (2003). Mistakes in Medical Ontologies: Where Do They Come From and How Can They Be Detected? In Pisanelli DM (ed) Ontologies in Medicine. In Proceedings of the Workshop on Medical Ontologies, 2003. IOS Press, Studies in Health Technology and Informatics, 2004, 145-164.
12. Noy, N., Sintek, M., Decker, S., Crubezy, M., Fergerson, R., Musen, M. (2000). Creating semantic web contents with Protégé-2000, IEEE Intelligent Systems.
13. Giunchiglia, F., Dutta, B., Maltese, V., Farazi, F. (2012). A facet-based methodology for the construction of a large-scale geospatial ontology.Journal of Data Semantics,1(1),57-73.
14. Mihalcea, R., Moldovan, D. I. (2001). Automatic generation of a coarse grained wordnet, NAACL Workshop on WordNet and Other Lexical Resources.

15. Soergel, D., Lauser, B., Liang, A., Fisseha, F., Keizer, J., Katz, S. (2004). Reengineering thesauri for new applications: the agrovoc example, Journal of Digital Information, 4.
16. Martinez-Cruz, C., Blanco, I., Vila, M. (2011). Ontologies versus relational databases: are they so different? A comparison. Artificial Intelligence Review, 1–20.
17. Jain, P., Hitzler, P., Yeh, P. Z., Verma, K., Sheth, A. P. (2010). Linked data is merely more data. Linked Data Meets Artificial Intelligence, pp. 82-86.
18. Berners-Lee, T., Hendler, J., Lassila, O. (2001). The Semantic Web. Scientific American.
19. Maltese, V., Farazi, F. (2013). A semantic schema for GeoNames. INSPIRE Conference.
20. Miller, G. A., Hristea, F. (2006). WordNet Nouns: classes and instances. Computational Linguistics, 32(1), 1-3.
21. Maltese, V. (2013). Imposing a semantic schema for the detection of potential mistakes in knowledge resources. 12th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE).
22. Hogan, A., Harth, A., Passant, A., Decker, S., Polleres, A. (2010). Weaving the pedantic web. In the 3rd International Workshop on Linked Data on the Web, WWW.
23. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U. (2008). Modular reuse of ontologies: Theory and practice. Journal of Artificial Intelligence Research, 31, 273–318.
24. A. Tawfik, F. Giunchiglia, V. Maltese (2014). A Collaborative Platform for Multilingual Ontology Development. World Academy of Science, Engineering and Technology, 8 (12).
25. F. Giunchiglia, B. Dutta, V. Maltese (2014). From Knowledge Organization to Knowledge Representation. Knowledge Organization. 41(1), 44-56.

Appendix: manual evaluation of the accuracy of the YAGO linking

The evaluation was conducted on 500 Wikipedia-WordNet pairs randomly selected and by incrementally computing the figures at blocks of 100 classes each. Final findings are summarized in Table 4. The first column shows the amount of classes analyzed. The second column shows the number of mistakes found in the corresponding block. The third column shows how the percentage of mistakes varies after each block of categories is analyzed. The final accuracy we found is 87%.

For instance, the class *Indoor arenas in Lithuania* is wrongly linked to the first WordNet sense of *arena* that is “a particular environment or walk of life”, while we believe that the correct one should be the third sense “a large structure for open-air sports or entertainments”.

However, as reported in the fourth and fifth columns, there are some cases in which, despite the proximity of the right sense, a more general (MG) or a more specific (MS) sense would be more appropriate. The last two columns show the percentage of mistakes updated taking into account such cases. Here the accuracy varies from 85% to 82%.

For instance, *Coal-fired power stations in Uzbekistan* is linked to *station* defined as “a facility equipped with special equipment and personnel for a particular purpose”, while a more appropriate class is clearly *power station* defined as “an electrical generating station”.

# classes	# mistakes	overall % mistakes	# MG senses	# MS senses	% mistakes MG	% mistakes MG + MS
100	11	0.11	2	1	0.13	0.14
200	7	0.09	2	3	0.11	0.13
300	12	0.10	0	2	0.11	0.13
400	16	0.12	2	3	0.13	0.15
500	20	0.13	4	3	0.15	0.18

Table 4. Manual evaluation of the YAGO linking

We also found 4 mistakes due to lack of senses in WordNet. For instance, *Eredivisie derbies* was mapped to the only sense of WordNet available for derby, i.e., "a felt hat that is round and hard with a narrow brim", while we believe that it refers more to football derby. They were not counted as mistakes in the table above. However, these are similar to the *argentine as fish* example given in the evaluation section.