deepschema.org: An Ontology for Typing Entities in the Web of Data

Panayiotis Smeros
EPFL, Switzerland
Panayiotis.Smeros@epfl.ch

Amit Gupta
EPFL, Switzerland
Amit.Gupta@epfl.ch

Michele Catasta
EPFL, Switzerland
Michele.Catasta@epfl.ch

Karl Aberer
EPFL, Switzerland
Karl.Aberer@epfl.ch

ABSTRACT

Discovering the appropriate type of an entity in the Web of Data is still considered an open challenge, given the complexity of the many tasks it entails. Among them, the most notable is the definition of a generic and cross-domain ontology. While the ontologies proposed in the past function mostly as schemata for knowledge bases of different sizes, an ontology for entity typing requires a rich, accurate and easily-traversable type hierarchy. Likewise, it is desirable that the hierarchy contains thousands of nodes and multiple levels, contrary to what a manually curated ontology can offer. Such level of detail is required to describe all the possible environments in which an entity exists in. Furthermore, the generation of the ontology must follow an automated fashion, combining the most widely used data sources and following the speed of the Web.

In this paper we propose deepschema.org, the first ontology that combines two well-known ontological resources, Wikidata and schema.org, to obtain a highly-accurate, generic type ontology which is at the same time a first-class citizen in the Web of Data. We describe the automated procedure we used for extracting a class hierarchy from Wikidata and analyze the main characteristics of this hierarchy. We also provide a novel technique for integrating the extracted hierarchy with schema.org, which exploits external dictionary corpora and is based on word embeddings. Finally, we present a crowdsourcing evaluation which showcases the three main aspects of our ontology, namely the accuracy, the traversability and the genericity. The outcome of this paper is published under the portal: http://deepschema.github.io.

CCS Concepts

•**Information systems** → *Information integration; Data extraction and integration;*

Keywords

Class Hierarchy, Taxonomy, Ontology, Wikidata, schema.org, Data Extraction, Data Integration, Entity Typing

LDOW '17. 3 April, 2017. Perth, WA, Australia.

1. INTRODUCTION

The definition of a generic and cross-domain ontology that describes all the types of the entities of the Web is considered as a very challenging task. In the past, many approaches that tried to address this problem proposed either manually curated ontologies or static schemata extracted from existed knowledge bases. However, both of these approaches have their deficiencies. A proper ontology for entity typing requires a rich, accurate and easily-traversable type hierarchy. Likewise, it is desirable that this hierarchy contains thousands of nodes and multiple levels. Such level of detail is required to describe all the possible environments in which an entity exists. Furthermore, the generation of the ontology must follow an automated fashion, combining the most widely used data sources and following the speed of the Web.

Currently, the most well-supported knowledge base and schema providers are Wikidata[1] and schema.org[2]. Wikidata is an initiative of Wikimedia Foundation for serving as the central repository for the structured data of its projects (e.g., for Wikipedia). Wikidata is also supported by Google which decided to shutdown its related project (Freebase[3]) in the middle of 2015 and since then has put a lot of effort on migrating the existing knowledge to Wikidata [10]. On the other hand, schema.org is an initiative of four sponsoring companies (Google, Microsoft, Yahoo and Yandex), supported by W3C as well, that aims on creating schemata that describe structured data on the web.

Both of these projects are trying to handle the plethora of heterogeneous, structured data that can be found on the web. Wikidata acts as a centralized data repository with a decentralized, community-controlled schema with millions of daily updates[4]. By contrast, schema.org proposes a very strict and rarely-updated schema, which is widely used by billions of pages across the web [9]. These two proposed approaches are considered complementary[5]. By bringing them

[1]http://wikidata.org
[2]http://schema.org
[3]http://freebase.com
[4]http://en.wikipedia.org/wiki/Wikipedia:Statistics
[5]http://meta.wikimedia.org/wiki/Wikidata/Notes/Schema.org_and_Wikidata

closer and unifying them, we form a rich, multi-level class hierarchy that can describe millions of entities in the Web of Data.

Such class hierarchy would be a very useful tool for many applications. For instance, in [15] the authors propose TRank, an algorithm for ranking entity types given an entity and its context. In the heart of TRank, a reference type hierarchy is traversed and the appropriate set of types for each entity is obtained. This type hierarchy combines information mostly from YAGO[6] and DBpedia[7]. However, none of these two data sources seems to suffice for the specific task of entity typing.

On one hand, YAGO's taxonomy inherits the class modeling of its sources (i.e., Wikipedia Categories[8] and WordNet[9]). Thus, nodes like *wikicat_People_murdered_in_British_Columbia* and *wordnet_person_100007846* are included in the taxonomy[10], making it inadequate to be traversed. DBpedia's ontology on the other hand, has a manually-curated and meaningful class hierarchy. Its volume though (only 685 classes) makes it inappropriate for describing accurately the millions of entities existing on the Web.

In another recent work, a knowledge graph named VoldemortKG was proposed [16]. VoldemortKG aggregates entities scattered across several web pages, which have both schema.org annotations and text anchors pointing to their Wikipedia page. Since entities are always accompanied by a schema, an ontology which contains the combined class hierarchy of the aforementioned data sources would complement this knowledge graph and increase its value.

In this paper we propose deepschema.org, the first ontology that combines two well-known ontological resources, Wikidata and schema.org, to obtain a highly-accurate, generic type ontology which is at the same time a first-class citizen in the Web of Data.

The main contributions of this paper are the following:

- the automated **extraction** procedure of the class hierarchy of Wikidata which is based on RDFS entailment rules
- the **analysis** of the main characteristics of this hierarchy, namely the structure, the instances, the language and the provenance
- the novel technique for the **integration** of the extracted hierarchy with schema.org which exploits external dictionary corpora and is based on word embeddings
- The crowd-sourced **evaluation** of the unified ontology which showcases the three main aspects of our ontology, namely the accuracy, the traversability and the genericity

The source code, the produced ontology and all the details about reproducing the results of this paper are published under the portal: http://deepschema.github.io.

[6]http://www.yago-knowledge.org
[7]http://dbpedia.org
[8]http://en.wikipedia.org/wiki/Wikipedia:Categorization
[9]http://wordnet.princeton.edu
[10]http://resources.mpi-inf.mpg.de/yago-naga/yago/download/yago/yagoTaxonomy.txt

The structure of the rest of the paper is organized as follows. In Section 2 we survey the related work while in Section 3 we provide more details on the Wikidata class hierarchy. In Section 4 we present the methods that we use for integrating Wikidata and schema.org. In Section 5 we describe the implementation and analyze the basic characteristics of the unified ontology. Finally, in Section 6 we evaluate the proposed methods and in Section 7 we conclude this work by discussing future directions.

2. RELATED WORK

The related work of this paper includes knowledge bases of general purpose, whose schemata comprise class hierarchical information as well as approaches that integrate such knowledge bases.

As mentioned above, Wikidata [17] is a community-based knowledge base i.e., users can collaboratively add and edit information. Wikidata is also multilingual, with the labels, aliases, and descriptions of its entities to be provided in more than 350 languages. A new dump of Wikidata is created every week and is distributed in JSON and experimentally in XML and RDF formats. All structured data from the main and the property namespace is available under the Creative Commons Public Domain Dedication License version 1.0 (CC0 1.0).

On the other hand, schema.org [5] provides a vocabulary which is widely used for annotating web pages and emails. This vocabulary is distributed in various formats (e.g., RDFa, Microdata and JSON-LD). The sponsors' copyrights in the schema are licensed to website publishers and other third parties under the Creative Commons Attribution-ShareAlike License version 3.0 (CC BY-SA 3.0).

The most well-know component of the LOD cloud is DBpedia [1]. It contains information which is automatically extracted mainly from the infobox tables of Wikipedia pages. Since it plays a central role in LOD cloud, DBpedia is the main hub in which many other datasets link to. The dataset is updated almost once every year, whereas there is also a live version [8] that continuously synchronizes DBpedia with Wikipedia. The data format that is used is the RDF and the publishing license is CC BY-SA 3.0.

Freebase [2] is a user-contributed knowledge base which integrates data from various data sources including Wikipedia and MusicBrainz. As stated before, Freebase has now shutdown and partially integrated to Wikidata. All the dumps of the dataset are published in the RDF format under the Creative Commons Attribution Generic version 2.5 (CC BY 2.5) license.

Another dataset that comprises information extracted from Wikipedia, WordNet and Geonames is YAGO [14]. The current version of YAGO has knowledge of more than 10 million entities (like persons, organizations, cities, etc.) assigned to more than $350,000$ classes. All the 4 dumps created by YAGO are distributed in the RDF and TSV data formats under the license CC BY-SA 3.0.

Wibi, the Wikipedia Bitaxonomy project [4] also induces a large-scale taxonomy for categories from the Wikipedia categories network. Wibi is based on the idea that information contained in Wikipedia pages is beneficial towards the construction of a taxonomy of categories and vice-versa. The most recent effort towards taxonomy induction over Wikipedia [6] proposes a unified taxonomy from Wikipedia with pages as leaves and categories as higher-level nodes us-

Axiomatic Triples		$(rdfs{:}subClassOf, rdfs{:}domain, rdfs{:}Class)$
		$(rdfs{:}subClassOf, rdfs{:}range, rdfs{:}Class)$
		$(rdf{:}type, rdfs{:}range, rdfs{:}Class)$
Entailment Rules	**rdfs2:**	$((A, rdfs{:}domain, B) \wedge (C, A, D)) \Rightarrow (C, rdf{:}type, B)$
	rdfs3:	$((A, rdfs{:}range, B) \wedge (C, A, D)) \Rightarrow (D, rdf{:}type, B)$
	rdfs9:	$((A, rdfs{:}subClassOf, B) \wedge (C, rdf{:}type, A)) \Rightarrow (C, rdf{:}type, B)$

Table 1: RDFS entailment rules for Extraction and Analysis of the Wikidata Class Hierarchy

classes	123,033
subclass relations	126,688
root classes	14,084
leaf classes	102,434
subgraphs	4,263
classes in big subgraph	118,120
subclasses in big subgraph	122,885
avg. subclasses per class	1.03
avg. depth of graph	7.93

Table 2: Statistics of the extracted class hierarchy

ing a novel set of high-precision heuristics.

Regarding the integration of such knowledge bases, many approaches have been proposed [12]. One interesting work that combines two of the aforementioned datasets is PARIS [13]. In this work the authors present some probabilistic techniques for the automatic alignment of ontologies not only in the instance but also in the schema level. The precision they achieve when they interconnect DBpedia and YAGO reaches 90%.

The authors of YAGO [14] also proposed a technique for constructing an augmented taxonomy, derived from Wikipedia and WordNet. The Wikipedia categories have a hierarchical structure which contains more thematic than ontological information (e.g., the category Football in France). Hence, the authors extract only the leaf categories, that semantically are closer to the notion of ontology classes. Then they align these categories with WordNet terms using string similarity methods which have precision of around 95%. Finally, they exploit the WordNet relation *hyponym* in order to construct the unified ontology.

The integration technique that we propose is based on word embeddings (Section 4) and despite its simplicity it discovers alignments with accuracy that is comparable to the one achieved by the two above methods (91%).

3. WIKIDATA

Wikidata is the main data source that we employ in deepschema.org. In this section we describe the methods for extracting a class hierarchy from Wikidata and we analyze the characteristics of this hierarchy. The described methods are not tightly coupled with a specific version of the data source, however, in the context of this paper we use Wikidata 20160208 JSON dump.

3.1 Class Hierarchy Extraction

The Wikidata JSON dump does not contain explicit information about the schema that accompanies the data. Every line of the dump consists of a unique entity and its attributes, described in a compact form[11]. The entities that represent classes and the entities that represent instances of classes are not distinguished in the dataset. Thus we have to apply semantic rules in order to extract them.

3.1.1 Semantic Rules

The rules that we apply to extract the taxonomy are based on the three axiomatic RDFS triples and the RDFS entailment rules 2 and 3 provided in Table 1. Intuitively, these rules imply that if X is of type Y, then Y is a class and if Z is a subclass of W, then Z and W are classes and the subclass relation holds between them.

Wikidata does not contain any *rdfs:subClassOf* or *rdf:type* properties, but it considers properties $P279$ (*subclass of*) and $P31$ (*instance of*) as equivalents of them (i.e., the have the same semantics). Hence we can apply the previous rules on these properties in order to extract the hierarchy.

3.1.2 Filtering Phase

As we will explain below, the raw form of the extracted hierarchy does not satisfy our requirements. Hence, we introduce a filtering phase in which we focus on two main aspects: i) the domain specific data sources and ii) the non-English labeled classes.

Domain Specific Data Sources. One of the main challenges for deepschema.org is genericity. Since data sources that apply to very narrow domains are imported to Wikidata, we introduce a filter in which we cleanse our hierarchy from such domain-specific information. As we discuss below, we had to drop more than the 75% of the extracted information in favor of keeping the hierarchy satisfyingly generic.

In order to track the provenance of the classes, we exploit the respective properties supported by Wikidata. The most widely-used provenance properties are the $P143$ (*imported from*) and the $P248$ (*stated in*). What we discovered

[11]More details about the data model of Wikidata can be found here: http://www.mediawiki.org/wiki/Wikibase/DataModel/Primer.

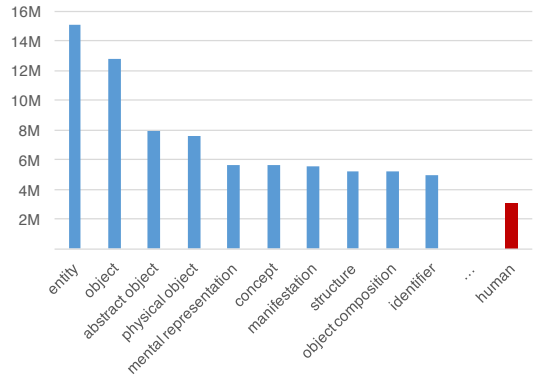

Figure 1: # of instances per Wikidata class

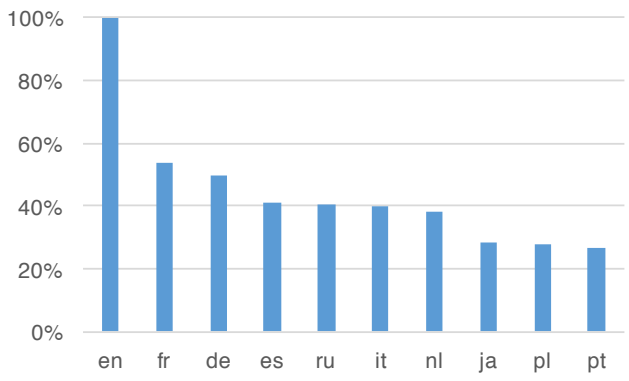

Figure 2: Coverage of label languages of Wikidata classes

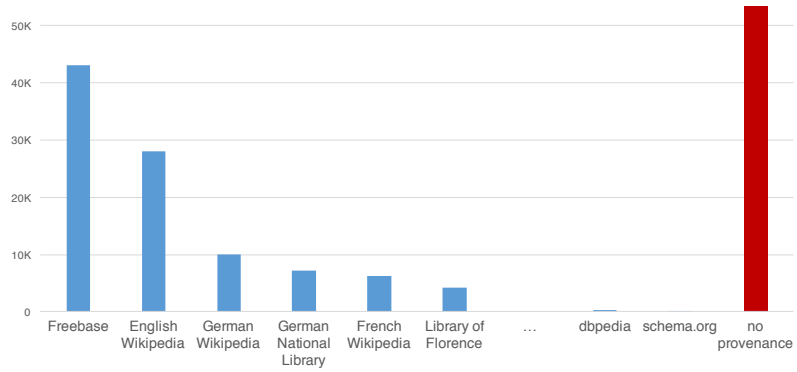

Figure 3: # of instances per Wikidata external contributor

is that many classes were imported to Wikidata from biological, chemical and mineral knowledge bases (e.g., NCBI[12], UniProt[13], Ensembl[14] and Mindat[15]). We consider these classes as very domain-specific, in terms of the objective of our hierarchy, and thus we apply a filter that prunes them.

Non-English Labeled Classes. Another filter that we apply is based on the language of the label of the extracted classes. As stated above, schema.org is expressed using the English language whereas Wikidata is multilingual. Including multilingual classes from Wikidata that do not contain English labels reduces tangibly the accuracy of our integration techniques (described in Section 4). Hence, we eliminate classes that do not fulfill this condition.

3.2 Analysis of the Class Hierarchy

Analyzing the characteristics of the extracted class hierarchy we focus on four main aspects: i) what is the structure of the hierarchy, ii) how the classes are populated with instances, iii) what is the distribution of their labels' language and iv) what is their provenance.

3.2.1 *Structure*

The overall statistics of the hierarchy are summarized in Table 2. The raw form of the extracted hierarchy includes 451,026 classes and 620,643 subclass relations. However, after the filtering phase we observe a decline of 73% in the classes and of 79% in the subclass relations. Nevertheless, the absolute size of the hierarchy is still considerably large (comparing it to other generic-purposed hierarchies), containing thousands of nodes which are distributed in multiple levels.

The classes of the hierarchy structure a graph which can be divided into 4,263 disconnected subgraphs. Most of these subgraphs have multiple roots with the total number of roots to be 14,084. Out of the subgraphs, one contains 96% of the total classes and 97% of the total subclass relations.

Since the vast majority (83%) of the classes are leaf classes, the average number of subclasses per class is very low (1.03). Furthermore, some parts of the graph are very flat, whereas some others are very deep, including multiple hierarchical levels, with the average depth of the graph to be 7.93.

3.2.2 *Instances*

A large amount of Wikidata classes are accompanied by instances. If a class contains instances, then these are inherited to all its superclasses because of the transitive property of the relation *subclass of*.

Based on this property and the RDFS entailment rule 9 (Table 1) and assuming that *P*31 (*instance of*) and *rdf:type* relations are equivalent, we managed to extract the in-

[12]http://www.ncbi.nlm.nih.gov
[13]http://www.uniprot.org
[14]http://www.ensembl.org
[15]http://www.mindat.org

stances, direct and inherited, of the Wikidata classes.

However, this approach does not discover all the underlying instances, because not all the existing classes are linked to their instances with the relation $P31$. For example, Quentin Tarantino's Wikidata entry[16] is *instance of* the class *Human*, whereas it is also connected with the class *film director* with the relation $P106$ (*occupation*). Hence, we observe that, subclasses of the class *Human* that denote occupation, are not as well-populated as their superclass.

On the other hand, if we try to add instances in classes independently of the relation that interconnects them, we include a lot of noise in the extracted hierarchy. In the same example, Quentin Tarantino would be an instance of the class *English* because he is connected with it by the relation P1413 (*languages spoken, written or signed*). One solution to this problem is to involve domain expert users to the procedure. These experts would decide or verify the relations that are eligible for interconnecting classes with instances (e.g., the relation *occupation*). However, the fact that our hierarchy contains thousands of classes and relations, deriving from many different domains, makes this solution inviable. Also this involvement would cancel the automated fashion in which we want to build our hierarchy.

Some interesting statistics about the instances of the Wikidata classes are presented in Figure 1 and summarized bellow:

- The class with the most instances is the class *Entity*[17] ($15M$ instances). *Entity*, as well as the following top-50 classes, is an abstract class, which means that most of its instances are inherited from subclasses based on the aforementioned rule.
- From the classes with direct instances, *Human* is the top one with $3M$ instances. What is remarkable is the fact that Wikidata uses the *Human* and not the *Person* class for people. *Person* is anything that can bear a personality, e.g. an artificial agent, etc.
- Other well-populated classes are the *Animal* class ($3M$ instances), the *Organization* class ($2.5M$ instances) including businesses, clubs and institutions, and the *Art Work* class ($1.5M$ instances) including music albums and movies.

3.2.3 Language

Wikidata follows a language-agnostic model according to which the identifiers of the entities intentionally consist of a character and a number. The multilingual support lies on the labels of these entities which are expressed in different languages. Currently, Wikidata contains entities in more than 350 languages [3].

In Figure 2 we can see the coverage of the label languages of the classes that were extracted from Wikidata. As we explained above, as a design choice we discard classes that do not have an English label. Thus the coverage of the English language is 100%. Interestingly enough, the next language is French with only 55% coverage. Thus, since English is the dominant language of our hierarchy, if we choose to export it in any other language (e.g., export only classes that have French label), we loose at least around one half of the information that we have acquired.

[16]http://www.wikidata.org/wiki/Q3772

[17]http://www.wikidata.org/wiki/Q35120

3.2.4 Provenance

Provenance information is very useful for crowdsourcing knowledge bases like Wikidata, because we can easily discard needless parts (as we did in the filtering phase above). As we can see in Figure 3 the main external contributor for the class hierarchy of Wikidata is Freebase with more than $40K$ classes. Then we have English Wikipedia with almost $30K$ classes, and Wikipedias and libraries in many other languages. DBpedia and schema.org have a few *equivalent class* links to Wikidata, whereas almost half of the classes do not comprise provenance information and thus we don't have a clue what is their source.

4. INTEGRATION WITH schema.org

In this section we describe the process of integrating the aforementioned Wikidata class hierarchy with the schema provided by schema.org. In the context of this paper we used 2.2 JSON release of schema.org.

We introduce several heuristics to perform the integration between Wikidata and schema.org (Figure 4). Each heuristic returns a candidate set of pairs of Wikidata nodes and schema.org nodes which are considered either as equivalent or the one as a subclass of the other. Heuristics use distributed vector representations of words computed by Glove [11] to compute a measure of similarity between words. The heuristics are described below:

- **Exact Match.** Maps a Wikidata node to a node in schema.org if they have the same labels. For example, the wikidata node with label "hospital" is mapped to schema.org node with label "Hospital".
- **Lemma Match.** Maps a Wikidata node to a node in schema.org if they have the same labels after lemmatization. WordNet [7] is used as a source for providing lemmatizations. For example, label "Cricket players" is converted after lemmatization into the label "cricket player". If the label of a node contains more than one word, then the node is lemmatized per token.
- **Single-word Similarity.** Maps a Wikidata node W to a schema.org node S if labels of both W and S have only one word and the cosine similarity between their glove vectors is greater than a fixed threshold (T_s). For example, Wikidata node with label "warehouse" is mapped to schema.org node with label "Store" because the cosine similarity between glove vectors for "warehouse" and "store" is greater than $T_s = 0.8$.
- **Exact Head Match.** Maps a Wikidata node to a schema.org node if the head[18] of the label of Wikidata node matches the label of schema.org node exactly or after lemmatization. For example, wikidata node with label "Kalapuyan languages" is mapped to schema.org as a subclass of the node with label "Language".
- **Head Similarity.** Maps a Wikidata mode to a schema.org node, if the cosine similarity between glove vectors of the heads of their labels is greater than T_s. For example, wikidata node with label "survey motor boat" is mapped to schema.org as a subclass of the

[18]Head is computed as the last token in the title, before "of" e.g., head of "national football leagues" is "leagues" and head of "national football leagues of south" is "leagues" as well.

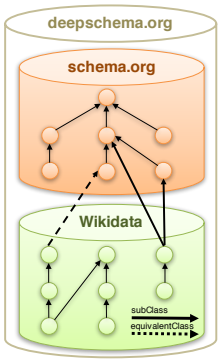

Figure 4: Structure of deepschema.org

Similarity threshold (T_s)	# of pairs
0.5	15112
0.6	8494
0.7	7329
0.8	6120
0.9	5586

Table 3: Output of the Integration phase

node with label "Vessel" based on the cosine similarity between "boat" and "vessel".

- **Instance Similarity.** Maps a Wikidata node W to a schema.org node S, if the average cosine similarity between the instances of W and the label of S is greater than T_s. This heuristic improves coverage of our approach by mapping nodes which would be otherwise unrelated based on their corresponding labels.

- **Subclass Similarity.** Similar to the previous heuristic, maps a Wikidata node W to a schema.org node S, if the average cosine similarity between the subclasses of W and the label of S is greater than T_s.

These heuristics result in pairs of Wikidata and schema.org nodes, which are mapped to each other. In Table 3 we can see the different number of pairs with respect to the different values of the threshold T_s.

5. IMPLEMENTATION

For the processing of the JSON dump of Wikidata and the extraction of the class hierarchy, we extended the Wikidata Toolkit[19] that is officially released and supported by Wikidata. In order to decrease the number of iterations through the dump, we follow a light-weight, in-memory approach in which we keep maps with the ids and the labels of the discovered classes and instances as well as with the relations among them. We also pipeline, where it is possible, the Extraction and Filtering phases. The user can choose the Wikidata dump which will be processed, turn on/off the various filters described above, and decide whether the output of the process will be in JSON, RDF or TSV format.

In order to analyze and compute various statistics about the hierarchy, we then process it as a graph using the Apache Spark GraphX library[20] and the various analytics functions that it supports.

For the integration step, we used Word2Vec[21] a two-layer neural network that processes text. We also downloaded and used the GloVe vectors trained from Wikipedia 2014 corpus and Gigaword corpus[22].

[19]https://github.com/Wikidata/Wikidata-Toolkit
[20]http://spark.apache.org/graphx
[21]http://deeplearning4j.org/word2vec
[22]http://nlp.stanford.edu/projects/glove

The code for all the tools that we used is open-source and can be found under the GitHub repository: http://github.com/deepschema/deepschema-toolkit.

Distribution. The license and the output format under which deepchema.org is distributed are described as follows:

- **License.** As mentioned in Section 2, Wikidata is distributed under the CC0 1.0 License and schema.org under the CC BY-SA 3.0 License. Since we combine the two datasets we chose to keep the most restrictive license. Thus, deepschema.org is distributed under the CC BY-SA 3.0 License.

- **Output Format.** deepschema.org is published under various formats (JSON, RDF and TSV) which are compatible with the most well-known ontology engineering tools (e.g., with Protégé[23]).

- **Releases.** Since deepschema.org is generated automatically, the tools described above can be, in principle, executed with any underline version of Wikidata and schema.org. Wikidata is updated weekly, whereas schema.org more rarely, thus, we can potentially release a new deepschema.org version every week.

6. EVALUATION

In this section we evaluate deepschema.org. Specifically, with the approach that we follow we focus on three main aspects of our ontology, namely i) the accuracy, ii) the traversability and iii) the genericity. The platform that we use in order to perform our crowdsourcing experiments is CrowdFlower[24].

6.1 Accuracy

In order to evaluate the accuracy we conducted a two-fold experiment. Both of the tasks of the experiment (which we describe in detail below) were designed to validate relations between classes. In the first task we validate internal relations within the employed data sources while in the second task we evaluate interlinks that we generated during the integration phase (Section 4). We asked around 100 people and the results were collated with majority voting (2 out of 3).

[23]http://protege.stanford.edu
[24]http://www.crowdflower.com

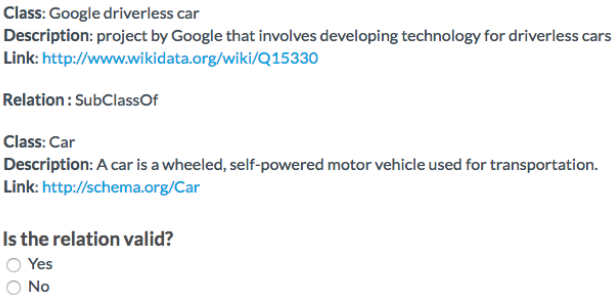

Figure 5: Crowdsourcing evaluation of the integration phase

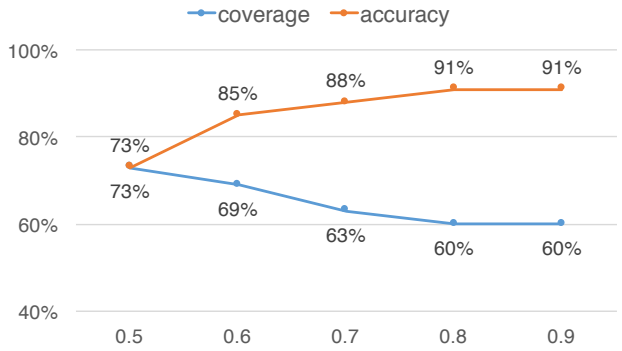

Figure 6: Coverage and Accuracy with respect to different thresholds

Each question is a multiple choice question in which we provide the classes to be connected, along with the suggested relation. Then, we request from the user to verify the correctness of the provided relation. In order to avoid ambiguities in the classes, we provide an additional description and a web link to each class. One example question that we asked the crowd can be shown in Figure 5 where we request the verification of the *subClassOf* relation for the classes *Google driveless car* from Wikidata and *Car* from schema.org.

6.1.1 Wikidata accuracy

On the first crowdsourcing task we assess the edge-level accuracy of the hierarchy we extracted from Wikidata. Since schema.org is generated and evaluated manually from domain experts, we consider it as 100% accurate and we do not involve the crowd for its assessment.

For Wikidata, we extracted at random 1000 edges, and asked the crowd if the relation between them (i.e., *subclassOf*) was meaningful. The reported accuracy that we obtained was 92%. In order to validate these results, we asked 3 ontology experts to evaluate part of the same 1000 edges. The average accuracy reported by the experts confirmed the results from our crowdsourcing task.

6.1.2 Integration accuracy

On the second task, we evaluated the output of the integration phase, as described in Section 4. The output consists of one class of Wikidata, one class of schema.org and the relation that we discovered between them. Since we used various similarity thresholds in the integration phase, we validated the accuracy for every one of them.

The results are summarized in Figure 6. As expected, while the threshold increases, the integration heuristics discover more accurate pairs of classes. For the thresholds 0.8 and 0.9 the accuracy reaches the 91%. The output of this experiment was also verified by the domain experts.

6.2 Traversability

In order to evaluate the traversability of deepschema.org we measure the amount of Wikidata leaf classes which have a direct path to the root of schema.org. Since schema.org has a tree structure, the problem is reduced to finding a path to any node of schema.org[25]. As we can see in Figure 6, lower similarity thresholds lead to the generation of more links and thus more paths which connect Wikidata and schema.org, at the expenses of accuracy.

The overall coverage we achieve is fairly low and this is explained mainly by the non-elegant structure of Wikidata. Both the schema and the instance information of Wikidata are controlled by the crowd and thus many of the classes that are not covered by schema.org are in fact noise (i.e., they are incorrectly annotated as classes or the *partOf* relation is mistakenly interpreted into the *subClassOf* relation). For example, the *List of NGC objects (5501-5750)*[26], which has been characterized as class, is actually a part of the *List of NGC objects*, which was imported to Wikidata from Wikipedia[27].

Furthermore, in some other cases, Wikidata classes were found to be more general and thus there was no actual superclass from schema.org to cover them besides the top classes like *Thing* (e.g., the class *Child Abuse*[28]). An easy workaround would be to connect every "orphan" Wikidata class to *Thing*. This would give us 100% coverage but it was out of the scope of the paper. Our goal was to construct deepschema.org with deep, traversable and meaningful paths at the cost of low coverage.

6.3 Genericity

Another goal for deepschema.org was to make it generic and applicable to multiple domains. One way to evaluate this characteristic was to employ a widely-used English dictionary and measure the coverage of its most frequent words that denote classes. In our experiment we used the Oxford 3000 subset of the Oxford English Dictionary[29].

The Oxford 3000 is a list of the 3000 most important English words. The keywords of the Oxford 3000 have been carefully selected by a group of language experts and experienced teachers as the words which should receive priority in vocabulary study because of their importance and usefulness. Despite its educational nature, Oxford 3000 gives

[25] By definition, there is always a unique path from every node of a tree to its root.
[26] http://www.wikidata.org/wiki/Q836200
[27] http://en.wikipedia.org/wiki/List_of_NGC_objects
[28] http://www.wikidata.org/wiki/Q167191
[29] http://www.oxfordlearnersdictionaries.com/wordlist/english/oxford3000

a good insight for the most commonly-used words in the English language.

Since this dictionary contains all the parts of speech (verbs, adjectives, etc.), and since classes are naturally described by nouns or noun phrases, we manually filtered the content of Oxford 3000 and kept only the words annotated as nouns and noun phrases.

The coverage of the filtered dictionary by deepschema.org is 81%. The latter confirms the generic nature and high coverage of our ontology.

7. CONCLUSIONS AND FUTURE WORK

In this paper we proposed deepschema.org, the first ontology that combines two well-known ontological resources, Wikidata and schema.org, to obtain a highly-accurate, generic type ontology which is at the same time a first-class citizen in the Web of Data. We described the automated procedure we used for extracting a class hierarchy from Wikidata and analyzed the main characteristics of this hierarchy. We also provided a novel technique for integrating the extracted hierarchy with schema.org, which exploits external dictionary corpora and is based on word embeddings. The overall accuracy of deepschema.org, reported by the crowdsourcing evaluation, is more than 90%, comparable to the accuracy of similar approaches that we have discussed in Section 2. Also, the evaluation of the traversability and the genericity showed very encouraging results by fulfilling the requirements that we had set up in the beginning.

Future work concentrates on employing more data sources as components of deepschema.org (e.g., Facebook's Open Graph[30]). By adding such data sources, deepschema.org will be established as the most generic and cross-domain class hierarchy. As we showcase in our evaluation, in spite of the filtering phase that we introduced, our ontology still contains a lot of noise. As future work we will extend these filters in order to further cleanse the noise that is imported mainly from Wikidata. Moreover, we will leverage the richness of the multilingual labels in Wikidata to produce versions of deepschema.org in multiple languages, although, as we have discussed in Section 3, the knowledge included in these versions will be limited. Finally, we will employ deepschema.org in real-world use-cases, like the one presented in [15], in which we will showcase the improvement obtained by the usage of our ontology.

8. ACKNOWLEDGEMENTS

This work was partially supported by the project *"Exploring the interdependence between scientific and public opinion on nutrition through large-scale semantic analysis"* from the Integrative Food Science and Nutrition Center (http://nutritioncenter.epfl.ch).

9. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A nucleus for a Web of open data. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4825 LNCS, pages 722–735, 2007.

[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data - SIGMOD '08*, page 1247, 2008.

[3] M. Färber, B. Ell, C. Menne, and A. Rettinger. A comparative survey of dbpedia, freebase, opencyc, wikidata, and yago. *Semantic Web Journal, July*, 2015.

[4] T. Flati, D. Vannella, T. Pasini, and R. Navigli. Two is bigger (and better) than one: the wikipedia bitaxonomy project. In *ACL (1)*, pages 945–955, 2014.

[5] R. Guha. Introducing schema. org: Search engines come together for a richer web. *Google Official Blog*, 2011.

[6] A. Gupta, F. Piccinno, M. Kozhevnikov, M. Paşca, and D. Pighin. Revisiting taxonomy induction over wikipedia. *COLING 2016*, 2016.

[7] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[8] M. Morsey, J. Lehmann, S. Auer, C. Stadler, and S. Hellmann. Dbpedia and the live extraction of structured data from wikipedia. *Program*, 46(2):157–181, 2012.

[9] P. Norvig. The semantic web and the semantics of the web: Where does meaning come from? In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 1–1, 2016.

[10] T. Pellissier Tanon, D. Vrandečić, S. Schaffert, T. Steiner, and L. Pintscher. From Freebase to Wikidata: The Great Migration. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1419–1428. International World Wide Web Conferences Steering Committee, 2016.

[11] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43, 2014.

[12] P. Shvaiko and J. Euzenat. Ontology matching: state of the art and future challenges. *IEEE Transactions on knowledge and data engineering*, 25(1):158–176, 2013.

[13] F. M. Suchanek, S. Abiteboul, and P. Senellart. PARIS : Probabilistic Alignment of Relations , Instances , and Schema. *Proceedings of the VLDB Endowment*, 5(3):157–168, 2011.

[14] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago. *Proceedings of the 16th international conference on World Wide Web - WWW '07*, page 697, 2007.

[15] A. Tonon, M. Catasta, G. Demartini, P. Cudré-Mauroux, and K. Aberer. TRank: Ranking entity types using the web of data. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8218 LNCS(PART 1):640–656, 2013.

[16] A. Tonon, V. Felder, D. E. Difallah, and P. Cudré-Mauroux. Voldemortkg: Mapping schema. org and web entities to linked open data. *Proceedings of the 15th International Semantic Web Conference*, 2016.

[17] D. Vrandečić and M. Krötzsch. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, Sept. 2014.

[30] http://ogp.me