Unsupervised Learning of an Extensive and Usable Taxonomy for DBpedia

Marco Fossati[♠] Dimitris Kontokostas[◇] Jens Lehmann[◇]

[♠]Fondazione Bruno Kessler
DKM Unit
Trento, Italy
fossati@fbk.eu

[◇]Universität Leipzig
AKSW Group
Leipzig, Germany
{kontokostas,lehmann}@informatik.uni-leipzig.de

ABSTRACT

In the digital era, Wikipedia represents a comprehensive cross-domain source of knowledge with millions of contributors. The DBpedia project transforms Wikipedia content into RDF and currently plays a crucial role in the Web of Data as a central multilingual interlinking hub. However, its main classification system depends on human curation, which causes it to lack coverage, resulting in a large amount of untyped resources. We present an unsupervised approach that automatically learns a taxonomy from the Wikipedia category system and extensively assigns types to DBpedia entities, through the combination of several interdisciplinary techniques. It provides a robust backbone for DBpedia knowledge and has the benefit of being easy to understand for end users. Crowdsourced online evaluations demonstrate that our strategy outperforms state-of-the-art approaches both in terms of coverage and intuitiveness.

Keywords

Wikipedia, Taxonomy Learning, Graph Algorithms, Natural Language Processing

1. INTRODUCTION

Wikipedia is the result of a crowdsourced effort and stands for the best digital materialization of encyclopedic human knowledge. Its data has been growingly drawing both research and industry interests, and has driven the creation of several knowledge bases, the most prominent being BabelNet [17], DBpedia [14], Freebase [3], YAGO [13], Wikidata [27], and WikiNet [15]. In particular, DBpedia[1] acts as the central component of the growing Linked Data cloud and benefits from a steadily increasing multilingual community of users and developers. Its stakeholders range from journalists [11] to governmental institutions [6], all the way to digital libraries [12]. The main contribution of DBpedia is to automatically extract structured data from unstructured (e.g., article abstracts) or semi-structured (e.g., infoboxes)[2] Wikipedia content.

The *de facto* model underpinning the classification of the encyclopedic entries, namely the DBpedia ontology (DBPO),[3] is maintained via a collaborative paradigm similar to Freebase. Any registered contributor can edit it by adding, deleting or modifying its content. The latest DBpedia release[4] contains 735 classes and 2,819 properties, which are highly heterogeneous in terms of granularity (cf. for instance the classes Band versus SambaSchool, both direct subclasses of Organisation) and are supposed to encapsulate the entire encyclopedic world. This indicates there is ample room to improve the quality of DBPO.

The Wikipedia category system is a fine-grained topical classification of Wikipedia articles, thus being natively suitable for encoding Wikipedia knowledge. DBpedia uses the category hierarchy as a supplementary classification system, while several taxonomization efforts such as [22, 23, 5, 9, 15, 16, 13], aim at mapping categories into types. However, their granularity is often very high, resulting in an arguably overly large set of items. From a practical perspective, it is vital to cluster resources into classes with intuitive labels, in order to simplify the end user's cognitive effort needed when querying the knowledge base. Hence, identifying a taxonomy based on a prominent subset of Wikipedia categories is a critical step to both extend and homogenize DBPO.

Furthermore, a clear problem of coverage has been recently pointed out [18, 1, 19, 10]. For instance, although the English Wikipedia contains about 4.9 million articles, DBpedia has only classified 2.8 million into DBPO. One of the major reasons is that a significant amount of Wikipedia entries does not contain an infobox, which is a valuable piece of information to infer the type of an entry. This results in a large number of untyped entities, thus restraining the exploitation of the knowledge base. Consequently, the extension of the DBpedia data coverage is a crucial step towards the release of richly structured and high quality data.

[1] http://dbpedia.org

[2] http://en.wikipedia.org/wiki/Help:Infobox

[3] http://mappings.dbpedia.org/server/ontology/classes/

[4] http://wiki.dbpedia.org/dbpedia-data-set-2015-04

Despite the number of similar initiatives, we argue that there is a need for a dataset with broad coverage and satisfactory intuitiveness. In this paper, we present *DBTax*, a completely data-driven methodology to automatically construct a comprehensive classification of DBpedia resources. Four features set DBTax apart from related approaches and constitute the main contributions of this paper:

1. Exhaustive type coverage over the whole knowledge base;
2. Focus on the actual usability of the schema from an end user's perspective;
3. Possibility of replication across different Wikipedia language chapters;
4. Fully unsupervised implementation, not requiring manual efforts for building annotated corpora.

The remainder of this paper is structured as follows. We first outline in Section 2 the problems we attempt to tackle and provide a high-level overview of the proposed solution. Section 3 contains our core contribution and illustrates in detail its major implementation phases. We corroborate our methodology with a report of its outcomes (Section 4), coverage comparisons with related resources, as well as an evaluation of both the taxonomy structure and the type assignment correctness (Section 5). In Section 6, we describe the policies to ensure access and sustainability of the output datasets. Finally, we review the state of the art in Section 7, before drawing our conclusions in Section 8.

2. PROBLEM STATEMENT

DBpedia resources are typed according to DBPO's classes. Nevertheless, a large amount of untyped resources is limiting the usability potential of the knowledge base. This is mainly due to the current classification paradigm described in [14], which heavily depends on Wikipedia infobox names and attributes in order to enable a manual mapping to DBPO. The availability and homogeneity of such semi-structured data in Wikipedia pages is unstable for two reasons, namely (a) the collaborative nature of the project, and (b) the linguistic and cultural discrepancies among all the language chapters. This results in several shortcomings, as highlighted in [10]. One major issue resides in the heterogeneous granularity of the ontology terms. Hence, some resources are typed with very specialized classes, while other similar entities have too generic types. Furthermore, resources can be wrongly classified, as a result of (a) the misuse of infoboxes by Wikipedia contributors, and (b) overlaps among the four mostly populated DBpedia classes, namely PLACE, PERSON, ORGANISATION and WORK.[5]

2.1 Prominent Nodes

As a solution to the aforementioned problems, we propose to automatically derive a taxonomy for the classification of DBpedia resources from a prominent subset of the Wikipedia category system, which provides a more reliable and almost complete knowledge backbone compared to infoboxes. We report below a high-level overview of our prominent node identification core algorithm, with the help of an example. A detailed description is provided in Section 3.2. The category with label MEDIA IN TRAVERSE CITY, MICHIGAN has 2 subcategories, namely (a) RADIO STATIONS IN TRAVERSE CITY, MICHIGAN (mentioned in 8 pages), and (b) TELEVISION STATIONS IN TRAVERSE CITY, MICHIGAN (mentioned in 4 pages). Both subcategories are leaf nodes. Thus, we make the parent category a *prominent node* and organize the 12 pages into a single cluster. Since this algorithm solely considers the category system structure, we incorporate linguistic processing and a usage-based technique. The former aims at simplifying the cluster label, which is renamed to MEDIA in our example. The latter weights the cluster depending on how often it is employed across all the Wikipedia language chapters.

3. GENERATING DBTAX

We envision the construction and the population of DBTax in four major stages:

1. Leaf node extraction;
2. Prominent node discovery;
3. Class taxonomy generation (T-Box);
4. Pages type assignment (A-Box).

First, we describe in Section 3.1 a method to identify initial leaf node candidates. In Section 3.2, we provide an overview of the prominent node discovery procedure step by step. The algorithms used to generate the class hierarchy are illustrated in Section 3.3. Finally, we assign types to Wikipedia pages (Section 3.4).

3.1 Stage 1: Leaf Nodes Extraction

The Wikipedia category system is organized in a cyclic graph data structure, which is of little use from a taxonomical perspective, due to its noisy nature. In fact, a class hierarchy best fits into a tree data structure, and we adopt a bottom-up approach to build it, starting from the leaves up to the root. Hence, the first stage takes as input the Wikipedia public database dumps[6] and outputs a set of *leaf nodes*, which is stored in a database table (i.e., NODE). Specifically, we use the Wikipedia tables encoding the links between the categories themselves, as well as between the categories and the pages. The procedure is implemented as follows: (a) we retrieve the full set of article pages, (b) we extract those categories that are linked to actual articles only, by looking up the outgoing links for each page, and out of them (c) we determine the set of categories with no subcategories.

3.2 Stage 2: Prominent Node Discovery

The following techniques are combined to identify the set of prominent category nodes:

1. *Algorithmic*, programmatically traversing the Wikipedia category system;

[5] `http://wiki.dbpedia.org/Datasets39/DatasetStatistics`

[6] `https://dumps.wikimedia.org`

2. *Linguistic*, identifying categories yielding is-a relations via Natural Language Processing;

3. *Multilingual*, leveraging interlanguage links.

The algorithmic technique is launched first and its output serves the other ones in a parallel fashion. We implement their outcomes in the form of attributes in the NODE database table, where a category represents a record.

3.2.1 Traversing the Leaf Graph

We now illustrate the procedure to programmatically process the Wikipedia category graph, starting from the set of leaf nodes produced in Section 3.1 and yielding a set of prominent node candidates. Its pseudocode is provided in Algorithm 1. The approach can be resumed as follows. Given as input a set of leaf nodes L, for each leaf l, we transitively traverse back to its set of parents P. For each such parent p, we check whether its set of children C is exclusively composed of leaves. If so, we consider p a prominent node and add it to the output set PN. Otherwise, we make l a prominent node. We use a boolean attribute to mark PN elements in the NODE table.

Algorithm 1 Prominent Node Discovery

Input: L **Output:** $PN \neq \varnothing$

```
PN ← ∅
for all l ∈ L do
    isProminent ← true; P ← getTransitiveParents(l)
    for all p ∈ P do
        C ← getChildren(p); areAllLeaves ← true
        for all c ∈ C do
            if c ∉ L then areAllLeaves ← false; break
        end for
        if areAllLeaves then
            PN ← PN ∪ {p}; isProminent ← false
    end for
    if isProminent then PN ← PN ∪ {l}
end for
return PN
```

3.2.2 NLP for is-a Relations

We adopt the approach applied in YAGO [13, 26] to identify prominent node candidates holding *is-a* relations. It relies on a straightforward yet powerful observation: since any Wikipedia category linguistically corresponds to a noun phrase, if its head appears in plural form, then that category is likely to be a conceptual one, and may serve as a class (cf. the paragraph on YAGO in Section 7). Specifically, we perform shallow syntactic parsing by means of the Noun Group Parser [25]. Categories are represented via link grammars [24], which are simple implementations of phrase structure grammars, the most complex being HPSG [21, 20].

For instance, Figure 1 explains how to parse the noun phrase (NP) PAST PRESIDENTS OF ITALY, which yields 3 chunks, namely a pre-modifier (PRE) PAST, a *head* PRESIDENTS and a post-modifier (POST) OF ITALY. We populate a new attribute of the NODE table with the head chunk. Afterwards, we exploit the Pling-Stemmer[7] to automatically mark prominent nodes having a plural head with a boolean attribute. The replicability of such method across multilingual Wikipedia deployments can be achieved via the following two strategies, each bearing its price: (a) exploitation of category interlanguage links (published by Wikipedia), at the cost of excluding categories with no English counterpart, and (b) language-specific implementations of the noun phrase parser and the stemmer, both at an intrinsic development expense and depending on the availability of language resources.

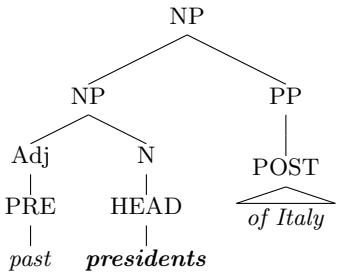

Figure 1: Example of a Wikipedia category phrase structure parsing tree

3.2.3 Interlanguage Links as a Weight

We leverage the LANGLINKS table of the Wikipedia database dumps to retrieve the number of interlanguage links for each prominent node candidate. This enables the implementation of a usage-driven weighting system, since we are able to induce a score assessing the usage of a given category among all the Wikipedia language editions. We populate a further attribute of the NODE table with the interlanguage links weight, and use it as a threshold to filter out underutilized items.

3.3 Stage 3: Class Taxonomy Generation

We reconstruct the full hierarchy of parent-child relations by recursively obtaining the set of parents for each leaf category, following a bottom-up direction.

3.3.1 Cycle Removal

The Wikipedia category graph contains cycles and so did the output of our first reconstruction attempt. In order to remove them and ensure a strict hierarchy, we apply Algorithm 2 in our processing pipeline. In brief, the algorithm traverses the graph in a breadth-first top-down fashion, starting from the root node (i.e., CONTENTS) and returns a tree T. For each node we encounter, we add it to T only if it has not been introduced yet. The set E keeps the already introduced nodes, while sets P and N keep the nodes for a specific tree level. The breadth-first approach for cycle removal favors shorter hierarchy paths: if a category exists in multiple levels of the graph, the node with the lowest depth will be added with a low distance to the root. However, we believe this choice both satisfies the goals of DBTax and complies with the philosophy of DBPO, namely to provide a high-level and general-purpose classification.

3.3.2 Pruning instances

The taxonomy we have obtained from the methods applied so far does not make the distinction between classes and instances. Thus, we need to leverage further post-processing to

[7] `http://resources.mpi-inf.mpg.de/yago-naga/javatools/doc/javatools/parsers/PlingStemmer.html`

Algorithm 2 Cycle Removal

Input: G **Output:** $T \neq \varnothing$

$T \leftarrow \varnothing$; $P \leftarrow getRootNode(G)$; $E \leftarrow P$
while $P \neq \varnothing$ **do**
  $N \leftarrow \varnothing$
  for all $p \in P$ **do**
    $C \leftarrow getChildren(p)$
    for all $c \in C$ **do**
      if $c \notin E$ **then**
        $E \leftarrow E \cup \{c\}$; $N \leftarrow N \cup \{c\}$; $T \leftarrow T \cup \{p, c\}$
    end for
  end for
  $P \leftarrow N$
end while
return T

prune instances and to produce a consumable resource. We opt for the name analysis approach proposed in [28], which assumes that instances are real-world entities. We leverage the DBpedia 3.9 release to filter out non-classes. Specifically, we combine the datasets containing labels, redirects and instances, and generate a list of labels for all DBpedia instances. By joining this list with the taxonomy, we managed to exclude 1,562 entries. Even though the pruning step cleaned DBTax from instances, it additionally removed many nodes from the hierarchy. This unavoidable side-effect partially decreased the quality of the T-Box. The reason is that nodes with pruned parents got attached directly to the root, thus resulting in broad paths (cf. Section 4).

3.4 Stage 4: Pages Type Assignment

We populate the taxonomy built in stage 3 by taking as input the heads of the prominent nodes returned in stage 2 and by leveraging the links between categories and Wikipedia article pages. In this way, we are able to assert an *instance-of* relation between a given page and the head of a category linked to that page. Once the type is assigned, its super and subtypes can be automatically inferred on account of the T-Box. We informally report below the foreseen procedure, which is applied to each prominent node head h.

1. Extract the set S of those categories having head $= h$;
2. Extract the pages linked to each category in S;
3. For each page p:
   (a) If it is an article page, then produce an assertion in the form of a triple $< p, \textit{instance-of}, h >$
   (b) If it is a category, recursively repeat from point 2 until the condition in point 3(a) is satisfied.

4. RESULTS

In order to enable the comparison across related resources, we process the same April 2013 English Wikipedia dumps as the DBpedia 3.9 release.[8] The outcomes of DBTax are three-fold, namely:

- The taxonomy (T-Box) automatically generated according to stage 3 (Section 3.3) is composed of 1,902 classes;
- 10,729,507 *instance-of* assertions (A-Box) are produced as output of stage 4 (Section 3.4). They are serialized into triples, according to the RDF data model.[9] We use the Turtle[10] syntax, which supports UTF-8-encoded International Resource Identifiers (IRIs), thus fitting well for multilingual Wikipedia pages with no need for escaping special characters. An example is reported as follows.

```
dbpedia:Combat_Rock a dbtax:Album .
```

- A total of 4,260,530 unique resources are assigned a type, 2,325,506 of which do not have one in the DBpedia 3.9 release.

5. EVALUATION

We use the following versions of the resources we compare to: (a) DBPO version 3.9;[11] (b) MENTA's underlying Wikipedia dumps date back to 2010; (c) SDType as per DBpedia version 3.9;[12] (d) YAGO types dataset as per DBpedia version 3.9;[13] (e) WiBi consumes the October 2012 English Wikipedia dump;[14] (f) Wikipedia categories from the same April 2013 English Wikipedia dumps; (g) Wikidata RDF exports from April 2014.[15] We decided to insert both MENTA and WiBi into our comparative evaluation anyway, since the former leverages knowledge from 271 languages, and the latter stands as the most recently published (2014) related approach. However, we recognize their performance might be relatively different on the April 2013 dump. Furthermore, the closest Wikidata dump we could access is one year newer. Hence, we expect a performance variation there as well. Finally, we could not retrieve the T-Box from MENTA and SDType, thus limiting their evaluation to the A-Box only. We could not build our experiments with Tìpalo [10], since the only available dataset[16] contains 547 unique entities, and has no overlap with our evaluation sets (cf. Section 5.2.1 and 5.3.1).

5.1 Coverage

Exhaustive type coverage over the whole knowledge base is a crucial objective in our contribution. We compute coverage as the number of resources for which at least one type is assigned, divided by the amount of actual Wikipedia article pages in the dump we process, excluding *redirect* pages. We report the values in Table 1. DBTax clearly outperforms all

[8] http://wiki.dbpedia.org/services-resources/datasets/data-set-39/dump-dates-39

[9] http://www.w3.org/TR/rdf11-concepts/

[10] http://www.w3.org/TR/turtle/

[11] http://downloads.dbpedia.org/3.9/dbpedia_3.9.owl.bz2

[12] http://downloads.dbpedia.org/3.9/en/instance_types_heuristic_en.ttl.bz2

[13] http://downloads.dbpedia.org/3.9/links/yago_types.ttl.bz2

[14] http://wibitaxonomy.org/wibi-ver1.0.tar.gz

[15] http://tools.wmflabs.org/wikidata-exports/rdf/exports/20140420/

[16] http://ontologydesignpatterns.org/ont/wikipedia/instance.rdf

the compared resources. Since our approach depends on the Wikipedia categories, one may object that articles with no assigned categories cannot be covered. However, at the time of writing this paper (August 2015), merely 2,263 English Wikipedia articles are uncategorized[17] (exclusively considering *content* categories, not *administrative* ones).[18] This corresponds to circa 0.045% of the total 4,934,195 articles.[19] Hence, the results we obtained for DBTax are in line with the statistics reported by the English Wikipedia. Moreover, DBTax identified 20.6% of DBPO manually curated classes, ranging from top-level (e.g., WORK), to deeply nested (e.g., BIOMOLECULE) ones. Such finding enables a natural mapping to DBPO.

Table 1: Type coverage of Wikipedia articles

Resource	Coverage
DBPO	.513
DBTax	**.994**
MENTA	.537
SDType	.147
YAGO	.673
WiBi	.794

5.2 T-Box Evaluation

We compare our results against DBPO, YAGO, WiBi, and Wikidata class hierarchies, as well as the Wikipedia category system itself, treating the Wikipedia categories as classes for the purpose of this evaluation only. We focus on (1) distinguishing classes from instances, and (2) hierarchy paths.

5.2.1 Task Anatomy

We pick a random sample of 50 classes from each resource and ask the evaluators the following questions: (a) *"Is this a class or an instance?"* (Class), and (b) *"Can this class be broken down into more than one class?"* (Breakable). For the hierarchy path evaluation, we pick a random sample of 50 *leaf* classes from each resource and generate the hierarchy path up to the root node (i.e., THING). We ask the evaluators the following questions: (a) *"Is this a valid class hierarchy path?"* (Valid), (b) *"Is this hierarchy too specific?"* (Specific), and (c) *"Is this hierarchy too broad?"* (Broad). The *Valid* question is meant to catch wrong hierarchies (e.g., THING ➤ CITY ➤ PLACE). The *Specific* and *Broad* questions aim at capturing such taxonomy design issues, although we recognize that they can be subjective and may depend on the use case. In fact, we expect a low agreement score, as we are assessing general-purpose taxonomies, with a high probability of cross-domain knowledge in our evaluation set. The *Breakable* and *Specific* questions involve leaf nodes only, while *Valid* is formulated with a path from a leaf node to the root. In total, 10 evaluators participated and each question was evaluated twice. The namespaces were hidden to avoid bias and the questions were globally randomized.

5.2.2 Discussion

[17] http://en.wikipedia.org/wiki/Category:All_uncategorized_pages

[18] http://en.wikipedia.org/wiki/Wikipedia:Categorization#Non-article_and_maintenance_categories

[19] http://meta.wikimedia.org/wiki/List_of_Wikipedias

Table 2: T-Box evaluation results. *C* is the ratio of classes in the taxonomy and *!Bre* the ratio of classes that cannot be broken into other classes. *V* is the ratio of valid hierarchy paths, *!S* the ratio of paths that are not too specific, and *!Bro* the ratio of paths that are not too broad

	C	!Bre	V	!S	!Bro
DBPO	.66	.67	.89	.97	.84
DBTax	.65	.76	.77	.98	.40
YAGO	.90	.38	.81	.55	.93
WiBi	.75	.38	.73	.41	.85
Wikidata	.19	.48	.85	.66	.88
Wikipedia	.81	.29	.66	.77	.78
Fleiss' κ	.32	.23	.23	.06	.30

Table 2 shows the overall results. Out of the four taxonomies, DBPO averagely performs slightly better. However, we expected such behavior, since it is a relatively small and manually curated ontology, compared to YAGO and DBTax. YAGO yields similar results to DBTax with respect to the *Valid* question. DBTax provides better non-breakable classes, as it solely consists of prominent nodes and does not create too specific hierarchies (cf. *!S*), as opposed to YAGO. Finally, DBTax stands last when it comes to broad hierarchies (cf. *!B*). This is due both to the cycle removal algorithm and especially to the instance pruning step (cf. Section 3.3), where several nodes were removed and leaf nodes got attached to the root. The main cause is the massive presence of instances in Wikipedia categories. The way we propose to overcome this is to outsource DBTax to the DBpedia ontology community and allow the community to perform the alignment. Although the *!Specific* and *!Broad* questions seem complementary, our intention is to additionally identify average hierarchy paths, suitable for a general-purpose taxonomy.

5.3 A-Box Evaluation

Assessing the actual usability of our knowledge base has the highest priority in our work. Moreover, estimating the quality of the assigned types must cope with subjectivity issues, as emphasized in [26]. Therefore, we decided to adopt an online evaluation approach with common users. Under this perspective, the major issue consists of gathering a sufficiently heterogeneous amount of judgments. Micro-payment services represent a suitable solution, since they allow us to outsource the evaluation task to a worldwide massive community of paid workers. We leverage the CrowdFlower platform,[20] which serves as a bridge to a plethora of crowdsourcing channels. In this way, we are able to simultaneously determine (a) the cognitive correctness of the assertions, and (b) the intuitiveness of the underlying semantics.

5.3.1 Task Anatomy

We randomly isolate 500 entities from those that do not have a type counterpart in DBpedia. Hence, we consider our evaluation set to be representative of the problem we are trying to tackle, namely to provide extensive classification coverage for DBpedia. While building our task, we aim at maximizing ease and atomicity. Workers are shown (1) a link to a Wikipedia page (i.e., the entity itself), labeled with

[20] http://www.crowdflower.com

the word *this* in the question *"What is this?"*, and (2) a type (i.e., the object of the instance-of relation, such as BAND), rendered in the form *"Is it a {type}?"*. Then, they are asked to (1) visit the page, and (2) judge whether the type is correct, by answering a *Yes/No* question.

For each entity, we elicit 5 judgments, thus gathering a total of 2,500. We prevent each worker from answering a question more than once by setting 500 maximum judgments per contributor and per IP. Finally, we ensure that all countries are allowed to work on our task and set the payment per page to $.03, where a page contains 5 entities. A cheating check mechanism is implemented via test questions, for which we supply the correct answer in advance. If a worker misses too many test answers within a given threshold (80% in our case), he or she will be banned and his or her *untrusted* judgments will be automatically discarded.

Table 3: Comparative A-Box evaluation on 500 randomly selected entities with no type coverage in DBpedia. ♠ indicates statistically significant difference with $p < .0005$ using χ^2 test, between DBTax and the marked resources

Resource	P	R	F_1	Agr	Untrusted
DBTax	.744	1	.853	.857	**518**
MENTA	.793	.589♠	.675	.826	1,093
SDType	.924	.098♠	.178	.899	1,723
YAGO	.461♠	.727	.565	.868	1,358
WiBi	.858	.597	.704♠	.924	2,075
Wikidata	.808	.982	.886	.913	1,847

5.3.2 Discussion

CrowdFlower provides a full report with detailed information for every single judgment made on the platform. For each question, an agreement score computed via majority vote weighted by worker trust is also included, and we calculate the average among the whole evaluation set. Table 3 displays the results obtained by processing the report. We compute precision as the ratio between positive answers and the total amount of answers, and recall as the ratio between positive answers and the sum of positive answers with the untyped entities (multiplied by 5 missing judgments). First, we notice that all resources are affected by recall issues, since they have a lack of type information, while our approach is always able to assign a type. This corroborates our findings on type coverage as per Table 1, where our system almost achieves 100%, in strong contrast to the other resources. To our surprise, DBTax also remarkably outperforms YAGO in terms of precision (validated by a statistical significance test), while the other resources generally behave better, although at a high recall cost. In a nutshell, DBTax scores satisfactorily high precision while reaching full recall. Via this trade-off, it achieves the best F_1 value, compared to automatically generated resources. Wikidata obtains the absolutely highest F_1, but we believe this might be due to the heavy manual curation efforts of millions of human contributors.[21]

Given similar agreement values (cf. the *Agr* column), the number of untrusted judgments may be viewed as a further indicator of the overall question ambiguity. In fact, we tried to maximize objectivity and simplicity when choosing test questions. However, it is known that the choice of taxonomical terms is always controversial, even for handcrafted taxonomies. Since the entities are identical in all the experiments, we can infer that the number of workers who missed the tests is directly influenced by the type ambiguity, which is the only variable parameter. In the light of the tangible discrepancy between the untrusted judgments values, we claim that DBTax is much more intuitive from a cognitive ergonomics perspective, even for common worldwide end users.

[21] https://www.wikidata.org/wiki/Special:Statistics

6. ACCESS AND SUSTAINABILITY

DBTax datasets will be included in the next and all subsequent official DBpedia releases. Within the release, it will serve as a complementary set of A- and T-Box statements to structure DBpedia resources. Thanks to the natural mapping to DBPO, an A-Box subset containing DBPO type assertions only is made available as well.[22] The first DBpedia release (v. 2015A) that will include this dataset is due on mid 2015. Since DBpedia is a pioneer in adopting and creating best practices for RDF publishing, being incorporated into its workflow guarantees regular updates. Long-term availability will be ensured through the DBpedia Association and the Leipzig Computing Data Center.

Until DBTax is not served by the regular DBpedia releases, the dataset is hosted at the *Italian DBpedia* chapter.[23] Moreover, it is registered on DataHub[24] and VOID metadata[25] is provided. Since DBTax is part of the official DBpedia releases, it benefits from the same users and developers communities, as well as support infrastructure.

7. RELATED WORK

The long strand of research focusing on automatic taxonomy learning from digital documents dates back to the 1970s [4]. It is out of scope for this paper to present an exhaustive literature review of such an extensive field of study. Instead, we concentrate on Wikipedia-related work. Ponzetto and Strube [22, 23] have pioneered the stream of the Wikipedia category system taxonomization efforts, providing a method for the extraction of a class hierarchy out of the category graph. While they integrate rule-based and lexico-syntactic-based approaches to infer intra-categories is-a relations, they do not distinguish between actual instances and classes.

Large-scale knowledge bases are experiencing a steadily growing commitment of both research and industry communities. A plethora of resources have been released in recent years. Table 4 reports an alphabetically ordered summary of the most influential examples, which all attempt to extract structured data from Wikipedia, although with different aims. BabelNet [17] is a multilingual lexico-semantic network, which recently moved towards a Linked Data compliant representation [7]. It provides wide-coverage lexicographic knowledge in 50 languages, where common concepts and real-world entities are linked together via semantic relations. Under this per-

[22] http://it.dbpedia.org/downloads/dbtax/A-Box-dbpo.nt.bz2

[23] http://it.dbpedia.org/downloads/dbtax/

[24] http://datahub.io/dataset/dbpedia-dbtax

[25] http://it.dbpedia.org/downloads/dbtax/void.ttl

spective, BabelNet emanates from the lexical databases community, with WordNet [8] being the most mature approach. In contrast to our work, priority is given to fine-grained conceptual completeness, rather than cognitively intuitive knowledge representation. DBpedia [14, 2] leads current approaches based on the automatic extraction of unstructured and semi-structured content from all the Wikipedia language chapters. It serves as the kernel of the Linked Data cloud, gathering a huge amount of research efforts in the Web of Data and Natural Language Processing. The underlying framework is strengthened by a vibrant open source community of users and developers. However, the current paradigm employed for the ontology weakens the data consumption capabilities. Freebase [3] is the result of a crowdsourced effort, bearing a fine-grained schema thanks to its contributors. Nevertheless, no type hierarchy exists: the collaborative paradigm has actually been privileged to logical consistency. Furthermore, multilingualism is biased towards English (cf. the ♢ symbol in Table 4), since information in other languages only appears when a Wikipedia page has an English counterpart. MENTA [5] is a massive lexical knowledge base, with data coming from 271 languages. The taxonomy extraction is carried out via supervised techniques, based on a manually annotated training phase, which diminishes the replicability potential, as opposed to our fully unsupervised method. Wikidata [27] stems from the Wikimedia Foundation and is the official Wikipedia sister project. Its data model differs from all the reviewed resources, since it favors plurality over authority, in a completely collaborative fashion. It builds upon claims instead of assertions, encapsulating both temporal and provenance aspects of a given fact. The schema is crowdsourced as in Freebase. WiBi [9]

Table 4: Overview of Wikipedia-powered knowledge bases (*C*ategories, *P*ages, *M*ultilingual, *3*rdparty data). ♢ indicates a caveat

Resource	C	P	M	3
BabelNet [17]	✔	✔	✔	✔
DBpedia [14, 2]	✔	✔	✔	✔
Freebase [3]	✘	✔	✔♢	✔
MENTA [5]	✔	✔	✔	✘
WiBi [9]	✔	✔	✘	✘
Wikidata [27]	✔	✔	✔	✔
WikiNet [15, 16]	✔	✔	✔	✘
WikiTaxonomy [22, 23]	✔	✘	✘	✘
YAGO [26, 13]	✔	✔	✘	✔

attempts to produce a double taxonomy by taking into account Wikipedia knowledge encoded both at the category and at the page layers. This is in clear contrast with our work, which concentrates on the category layer to construct a classification backbone for the page layer. Similarly to us, it does not leverage third party resources and is implemented under an unsupervised paradigm. WikiNet [15, 16] is built on top of heuristics formulated upon the analysis of Wikipedia content to deliver a multilingual semantic network. Besides is-a relations, like we do, it also learns other kinds of relations. While it seems to attain wide coverage, a comparative evaluation performed in [9] highlights very low precision.

The approach that most influenced our work is YAGO [13, 26]. Its main purpose is to provide a linkage facility between categories and WordNet terms. Conceptual categories (e.g. PERSONAL WEAPONS) serve as class candidates and are separated from administrative (e.g. CATEGORIES REQUIRING DIFFUSION), relational (e.g. 1944 DEATHS) and topical (e.g. MEDICINE) ones. Similarly to us, linguistic-based processing is applied to isolate conceptual categories.

On the other hand, the recently proposed automatic methods for type inference [18, 1, 19, 10] have yielded resources that may enrich, cleanse or be aligned to DBPO's class hierarchy. Moreover, they can serve as an assisting tool to prevent redundancy, namely to alert a human contributor when he or she is trying to add some new class that already exists or has a similar name. Hence, these efforts represent alternative solutions compared to our work, with Tìpalo [10] being the most related one.

8. CONCLUSION

DBTax is the outcome of a completely data-driven approach to convert the chaotic Wikipedia category system into an extensive general-purpose taxonomy. As a result of our four-step processing pipeline, we generated a hierarchy of 1,902 classes and automatically assigned types to roughly 4.2 million DBpedia resources. Thus, we provide a significant coverage leap, as opposed to DBpedia (with only 2.2 million typed resources) and to related automatic approaches. Moreover, online evaluations in a crowdsourcing environment demonstrate that DBTax is not only comparable to the manually curated DBpedia ontology (DBPO) in terms of taxonomical structure, but is also outstandingly intuitive for common end users, while achieving the best precision and recall trade-off. DBTax is currently deployed in the Italian DBpedia chapter SPARQL endpoint[26] and will be included in all future DBpedia releases.

We envision DBTax to serve as a balance between DBPO and YAGO, as we argue that DBPO is very limiting and YAGO far too large for real-world use cases. For future work, we plan to merge the T-Box into the DBpedia mappings wiki[27] and allow the DBpedia community to further curate and organize it. We believe this will also cater for the broad hierarchy paths that resulted from the pruning steps. Furthermore, a word sense disambiguation technique is scheduled for implementation, in order to distinguish between homonymous classes. Since the A-Box may state multiple heterogeneous types for a resource (e.g., ELVIS PRESLEY is both a SINGER and a PROTESTANT), we foresee to rank types according to their statistical relevance, such as the absolute frequency of instances. Finally, we expect to additionally exploit the Wikipedia category interlanguage links, in order to (a) produce multilingual labels for DBTax, (b) pinpoint additional classes that our process did not extract in English, and (c) deploy the approach to DBpedia language chapters besides English and Italian, at the price of excluding categories with no English counterpart.

[26] http://it.dbpedia.org/2015/02/dbpedia-italiana-release-3-4-wikidata-e-dbtax/?lang=en

[27] http://mappings.dbpedia.org

Acknowledgments

This work was partially supported by Google via the Google Summer of Code program and by the European Union's 7th Framework & H2020 programs via the GeoKnow (GA no. 318159) and the ALIGNED (GA no. 644055) projects.

9. REFERENCES

[1] A. P. Aprosio, C. Giuliano, and A. Lavelli. Towards an automatic creation of localized versions of dbpedia. In *The Semantic Web–ISWC 2013*, pages 494–509. Springer, 2013.

[2] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.

[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the international conference on Management of data*, pages 1247–1250. ACM, 2008.

[4] N. Calzolari, L. Pecchia, and A. Zampolli. Working on the italian machine dictionary: a semantic approach. In *Proceedings of the 5th Conference on Computational Linguistics*, volume 2, pages 49–52. Association for Computational Linguistics, 1973.

[5] G. de Melo and G. Weikum. Menta: Inducing multilingual taxonomies from wikipedia. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 1099–1108. ACM, 2010.

[6] L. Ding, T. Lebo, J. S. Erickson, D. DiFranzo, G. T. Williams, X. Li, J. Michaelis, A. Graves, J. G. Zheng, Z. Shangguan, et al. Twc logd: A portal for linked open government data ecosystems. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(3):325–333, 2011.

[7] M. Ehrmann, F. Cecconi, D. Vannella, J. McCrae, P. Cimiano, and R. Navigli. Representing multilingual data as linked data: the case of babelnet 2.0. In *Proceedings of the 9th Language Resources and Evaluation Conference*, 2014.

[8] C. Fellbaum. *Wordnet: an Electronic Lexical Database*. MIT Press Cambridge, 1998.

[9] T. Flati, D. Vannella, T. Pasini, and R. Navigli. Two is bigger (and better) than one: the wikipedia bitaxonomy project. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014.

[10] A. Gangemi, A. G. Nuzzolese, V. Presutti, F. Draicchio, A. Musetti, and P. Ciancarini. Automatic typing of dbpedia entities. In *The Semantic Web–ISWC 2012*, pages 65–81. Springer, 2012.

[11] J. Gray, L. Chambers, and L. Bounegru. *The Data Journalism Handbook*. O'Reilly Media, Inc., 2012.

[12] B. Haslhofer, E. Momeni, M. Gay, and R. Simon. Augmenting europeana content with linked data resources. In *Proceedings of the 6th International Conference on Semantic Systems*, page 40. ACM, 2010.

[13] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. Yago2: a spatially and temporally enhanced knowledge base from wikipedia. *AI*, 194:28–61, 2013.

[14] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195, 2015.

[15] V. Nastase and M. Strube. Transforming wikipedia into a large scale multilingual concept network. *Artificial Intelligence*, 194:62–85, 2013.

[16] V. Nastase, M. Strube, B. Boerschinger, C. Zirn, and A. Elghafari. Wikinet: A very large scale multi-lingual concept network. In *Proceedings of the 5th Language Resources and Evaluation Conference*, 2010.

[17] R. Navigli and S. P. Ponzetto. Babelnet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *AI*, 193:217–250, 2012.

[18] H. Paulheim and C. Bizer. Type inference on noisy rdf data. In *The Semantic Web–ISWC 2013*, pages 510–525. Springer, 2013.

[19] A. Pohl. Classifying the wikipedia articles into the opencyc taxonomy. In *Proceedings of the Web of Linked Entities Workshop in conjuction with the 11th International Semantic Web Conference*, 2012.

[20] C. Pollard and I. A. sag. *Information-based Syntax and Semantics: vol. 1: Fundamentals*. Center for the Study of Language and Information, Stanford, CA, USA, 1988.

[21] C. Pollard and I. A. sag. *Head-driven Phrase Structure Grammar*. University of Chicago Press, 1994.

[22] S. P. Ponzetto and M. Strube. Deriving a large scale taxonomy from wikipedia. In *Proceeding of AAAI*, volume 7, pages 1440–1445, 2007.

[23] S. P. Ponzetto and M. Strube. Taxonomy induction based on a collaboratively built knowledge repository. *Artificial Intelligence*, 175(9-10):1737–1756, 2011.

[24] D. D. Sleator and D. Temperley. Parsing english with a link grammar. *CoRR*, abs/cmp-lg/9508004, 1995.

[25] F. M. Suchanek, G. Ifrim, and G. Weikum. Leila: Learning to extract information by linguistic analysis. In *Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 18–25, 2006.

[26] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, pages 697–706. ACM, 2007.

[27] D. Vrandečić and M. Krötzsch. Wikidata: a free collaborative knowledge base. *Communications of the ACM*, 57(10):78–85, 2014.

[28] C. Zirn, V. Nastase, and M. Strube. Distinguishing between instances and classes in the wikipedia taxonomy. In *Proceedings of the 5th European Semantic Web Conference*, pages 376–387, 2008.