Turning the DBpedia Ontology Upside Down with DBTax

Marco Fossati[1], Dimitris Kontokostas[2], and Jens Lehmann[2]

[1] Fondazione Bruno Kessler, Web of Data Unit, Trento, Italy
fossati@fbk.eu
[2] Universität Leipzig, Institut für Informatik, AKSW, Leipzig, Germany
{lastname}@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 aims at transforming Wikipedia content into structured intelligence via the Linked Open Data principles and currently plays a crucial role in the development of the Web of Data as a central interlinking hub. We present DBTax, an approach to automatically derive a taxonomy from the Wikipedia category system, through the combination of several interdisciplinary techniques. Not only it provides a robust backbone for DBpedia knowledge, but it is also an ideal candidate to enrich the handcrafted DBpedia class hierarchy. Results demonstrate that DBTax outperforms state-of-the-art resources both in terms of coverage and quality.

Keywords: Wikipedia, Taxonomy Derivation, Graph Algorithms, Natural Language Processing

1 Introduction

A major portion of the World Wide Web content is nowadays represented as unstructured data, namely documents. Understanding its meaning is a complex task for machines and still relies on subjective human interpretations. The Web of Data envisions its evolution as a repository of machine-readable structured data. This would enable an automatic and unambiguous content analysis and its direct delivery to end users. The idea has not only engaged a long strand of research, but has also been absorbed by the biggest web industry players. Companies such as Google, Facebook and Microsoft, have already adopted large-scale semantics-driven systems, namely Google's *Knowledge Graph*,[3] Facebook's *Graph Search*,[4] and Microsoft's *Satori*.[5] Moreover, the World Wide Web Consortium, together with the Linked Data[6] initiative, has provided a standardized technology stack to publish freely accessible interconnected datasets.

[3] http://www.google.com/insidesearch/features/search/knowledge.html
[4] https://www.facebook.com/about/graphsearch
[5] http://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing/
[6] http://linkeddata.org

On the other hand, Wikipedia is the result of a crowdsourced effort and stands for the best digital approximation 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 [16], DBpedia [12], Freebase [3], YAGO [11], and WikiNet [14]. In particular, DBpedia[7] 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 [9] to governmental institutions [5], all the way to digital libraries [10]. The main purpose of DBpedia is to automatically extract structured data from unstructured (e.g. article abstracts) or semi-structured (e.g. infoboxes[8]) Wikipedia content.

The model underpinning the classification of the encyclopedic entries, namely the DBpedia ontology,[9] is maintained via a collaborative paradigm similar to Freebase. Any registered contributor can edit it by adding, deleting or modifying its content. At the time of the writing, it contains 649 classes and 2782 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. Therefore, there is still ample room to improve the quality of the knowledge encoded into DBpedia.

The Wikipedia category system is a fine-grained topical classification of Wikipedia articles, thus being natively suitable for encoding Wikipedia knowledge. Efforts such as YAGO [11] aim at mapping those categories into types. However, their granularity is often too high and the resulting set of classes is too large. From a practical perspective, it is vital to cluster resources into classes that are intuitive at query time, namely when end users want to ask questions to the knowledge base. Hence, identifying a taxonomy based on a prominent subset of Wikipedia categories is a critical step to both extend the ontology coverage and prune too specific classes. Furthermore, a clear problem of coverage has been recently pointed out in [18,1,19,17]. For instance, although the English Wikipedia contains about 4 million pages, DBpedia has only classified 2.2 million pages into its ontology. One of the major reasons is that a significant amount of Wikipedia articles do not contain an infobox, which is a valuable piece of information to infer the type of an article. 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.

In this paper, we present *DBTax*, a completely data-driven methodology to automatically construct a consistent classification of DBpedia resources and to provide exhaustive type coverage. Four features set DBTax apart from related approaches and constitute the main contributions of this paper:

1. Focus on an optimal trade-off between knowledge granularity and ontology usability for effective knowledge base querying.

[7] `http://dbpedia.org`
[8] `http://en.wikipedia.org/wiki/Help:Infobox`
[9] `http://mappings.dbpedia.org/server/ontology/classes/`

2. Replicability across Wikipedia deployments, particularly suitable for different language chapters. Hence, multilingualism is achieved.
3. Integration with DBpedia open source developments, which are backed by a user community, as opposed to the publication of a standalone resource.
4. Fully automatic implementation, not requiring human intervention.

The remainder of this paper is structured as follows. Section 2 outlines the problems we attempt to tackle and provides a high-level overview of the proposed solution. Next, we briefly describe in Section 3 the Wikipedia category system with an emphasis on its usage in English articles. Section 4 contains our core contribution and illustrates in detail its major implementation phases. We corroborate our methodology with a report of its outcomes, coverage comparisons with DBpedia and YAGO, as well as an evaluation of both the taxonomy structure and the type assignment correctness (Section 5). Finally, we review the state of the art in Section 6, before drawing our conclusions in Section 7.

2 Revolutionizing the DBpedia Classification

DBpedia resources are typed according to the DBpedia ontology class hierarchy. 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 [12], which heavily depends on Wikipedia infobox names and attributes in order to enable a manual mapping to the DBpedia ontology. The availability and homogeneity of such semi-structured data in Wikipedia pages is unstable, owing to 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 [17]. 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.

As a solution to the above mentioned 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 idea, with the help of an example.

The category 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 in the graph. Thus, we make the parent category a *prominent node* and organize the 12 pages into a single cluster. The cluster label is finally renamed to Media, on account of a

natural language processing step. A detailed description of the supplementary techniques we implement is provided in Section 4.

3 The Wikipedia Category System

The Wikipedia category system is the primary source of classification in Wikipedia. It contains category pages that list the articles belonging to a specific category (e.g., `Elvis Presley` is listed under AMERICAN ROCK SINGERS). In addition, it provides navigational links to all the Wikipedia pages in the form of a hierarchy. In this way, readers can browse and find articles by only looking at their essential characteristics.

Wikipedia categories are organized in a graph, eventually containing cycles. The graph has links between a given Wikipedia page to its related topics, but is of little use from a taxonomical perspective, due to its noisy nature. Typically, cycles and non-conceptual categories must be filtered out. However, leaf categories can be leveraged as the starting point to drive the construction of a meaningful taxonomy. A *leaf category* is defined as a Wikipedia category without outgoing links to any other category. Wikimedia exposes public database dumps[10] which encode the links between the categories themselves, as well as between the categories and the pages. Thus, we leveraged them as our main input source.

Categories Usage. The PAGE database table contains not only actual articles, but also other resources such as *talk* and *template* pages. Hence, we need a method to filter out unnecessary pages. As per June 2013, 726,184 categories have no links to Wikipedia pages. On the other hand, 693 have more than 10,000 pages, and are possibly administrative categories or higher nodes of the category graph. The bootstrap elements of our prominent nodes discovery methodology (cf. Section 4) reside in the set of categories not having any subcategories. We consider that these categories represent the leaf nodes of the whole category system.

4 Generating DBTax

We envision the construction and the population of our *is-a* relations class hierarchy in four major stages:

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

First, we describe in Section 4.1 a method to identify initial leaf node candidates. In Section 4.2, we provide an overview of the prominent node discovery procedure step by step. The algorithms used to generate the class hierarchy (T-Box) are illustrated in Section 4.3. Finally, we infer types for Wikipedia pages (Section 4.4).

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

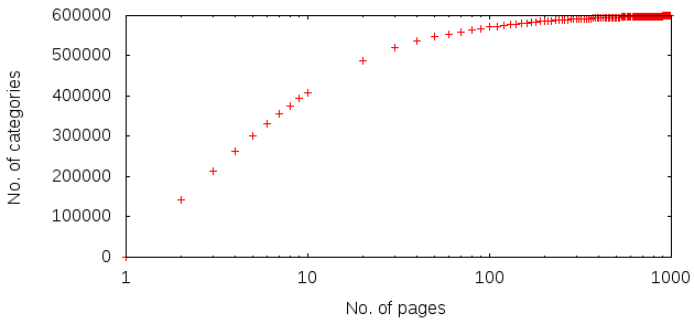

Fig. 1. Category usage in Wikipedia pages

4.1 Stage 1: Leaf Nodes Selection

An optimal trade-off between knowledge granularity and query agility is a critical aspect for the construction of our data-driven classification schema. Therefore, we decided to select a subset of leaf nodes according to their actual usage, which we assume not to be affected by radical changes over time. To achieve this, we need to (a) retrieve the full set of article pages, (b) extract those categories which are linked to actual articles only, by looking up the outgoing links for each page, and (c) plot a category distribution graph.

Figure 1 illustrates a portion of the category distribution curve. Specifically, it displays the amount of categories with respect to the number of pages mentioning them. First, we want to discard unused categories, namely when they are not referenced in any page. Next, we notice that roughly 150,000 categories appear in 2 pages. Initially, the curve seems to grow exponentially until the number of pages reaches a value of 60. After that, it stabilizes in a range between 555,000 and 600,000. In fact, for $number\,of\,pages \to \infty$, the $number\,of\,categories$ values reach a limit equaling to 603,423. Therefore, the curve trend approximates to an asymptotic logarithmic function, i.e., $\log(number\,of\,pages)$. We formalize this as follows:

$$x := number\,of\,pages, \quad \varphi(x) := number\,of\,categories$$
$$\varphi : \mathbb{N}^+ \to \mathbb{N}_0$$
$$x \mapsto \varphi(x)$$
$$\varphi(x) \sim \log(x), \quad \lim_{x\to\infty} \varphi(x) = 603,423$$

The reader may refer to Table 1 for a list of significant value pairs.

We consider the interval $0 < x < 90$, which we estimate to be optimal for cutting out administrative categories, on account of manual assessments. Finally,

Table 1. Category usage in numbers

Pages	2	4	60	**90**	100	500	990
Categories	141,888	262,363	554,155	**568,400**	571,286	595,577	599,029

the included leaf nodes set is stored in a database table, i.e., NODE, while the excluded one is assumed to be composed of non-conceptual categories.

4.2 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
2. *Linguistic*, identifying categories yielding is-a relations via Natural Language Processing
3. *Multilingual*, leveraging interlanguage links
4. *Post-mortem*, further filtering irrelevant items through Freebase

We implement the outcomes of each technique in the form of attributes in the NODE database table, where a category represents a record.

Traversing the Leaf Graph. We now illustrate the procedure implemented to programmatically process the Wikipedia category graph subset selected in Section 4.1. Its pseudocode is provided in Algorithm 1(a). The approach can be resumed as follows. Given as input the set of leaf nodes L produced in stage 1, for each leaf l, we traverse back to its set of parents P. We check whether the set of children C of every parent p are 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.

NLP for is-a Relations. We adopt the approach applied in YAGO [11,26] to identify category 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. Specifically, we perform shallow syntactic parsing over prominent node names, 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 [20,21]. For instance, Figure 2 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[11] to automatically mark prominent nodes having a plural head with a boolean attribute. Although morphosyntactic processing requires language-specific implementations, multilingualism can be ensured through the exploitation of category interlanguage links, published by Wikipedia.

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

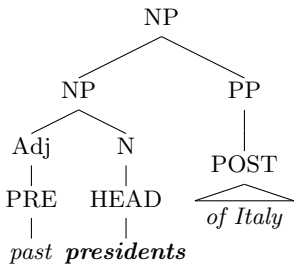

Fig. 2. Example of a Wikipedia category phrase structure parsing tree

Interlanguage Links as a Weight. We leverage the LANGLINKS table of the Wikipedia database dumps to retrieve the number of interlanguage links for each category 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.

Post-processing Filter. Since heads are linked to categories in a 1-n relation, they serve as a form of clustering prominent nodes. However, we notice that the result set includes a large number of real-world entities, which are not suitable to represent classes in the final taxonomy, but rather instances. Freebase serves here as a further filter, since it contains a huge amount of entities. Hence, we extract lists of entries from the top-4 most frequent types, namely person, location, organization, and recordings.[12] These lists are intersected with our result set in order to produce the final head candidate set.

4.3 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.

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 applied Algorithm 1(b) in our processing pipeline. In brief, the algorithm traverses the graph in a breath-first fashion, starting from the root node (i.e., `Thing`) 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.

Pruning instances. The taxonomy we obtain from the methods applied so far does not make the distinction between classes and instances. Thus, we need to leverage further post-processing to prune instances and produce a consumable

[12] The Freebase schema is available at `http://www.freebase.com/schema`

Algorithm 1 **(a)** Prominent Node Discovery, **(b)** Cycle Removal

Input: L
Output: $PN \neq \varnothing$
1: $PN \leftarrow \varnothing$
2: **for all** $l \in L$ **do**
3: $isProminent \leftarrow$ **true**
4: $P \leftarrow getParents(l)$
5: **for all** $p \in P$ **do**
6: $C \leftarrow getChildren(p)$
7: $areAllLeaves \leftarrow$ **true**
8: **for all** $c \in C$ **do**
9: **if** $c \in L$ **then**
10: $areAllLeaves \leftarrow$ **false**
11: **break**
12: **end if**
13: **end for**
14: **if** $areAllLeaves$ **then**
15: $PN \leftarrow PN \cup \{p\}$
16: $isProminent \leftarrow$ **false**
17: **end if**
18: **end for**
19: **if** $isProminent$ **then**
20: $PN \leftarrow PN \cup \{l\}$
21: **end if**
22: **end for**
23: **return** PN

Input: G
Output: $T \neq \varnothing$
1: $T \leftarrow \varnothing$
2: $P \leftarrow getRootNode(G)$
3: $E \leftarrow P$
4: **while** $P \neq \varnothing$ **do**
5: $N \leftarrow \varnothing$
6: **for all** $p \in P$ **do**
7: $C \leftarrow getChildren(p)$
8: **for all** $c \in C$ **do**
9: **if** $c \notin E$ **then**
10: $E \leftarrow E \cup \{c\}$
11: $N \leftarrow N \cup \{c\}$
12: $T \leftarrow N \cup \{p, c\}$
13: **end if**
14: **end for**
15: **end for**
16: $P \leftarrow N$
17: **end while**
18: **return** T

taxonomy. We adopt the name analysis approach proposed in [27], 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 DBTax, we managed to exclude 1,562 entries. Even though this step cleaned DBTax from instances, it also removed many nodes in the hierarchy and resulted in attaching their corresponding leaf nodes directly to the root node (cf. Section 5).

4.4 Stage 4: Pages Type Inference

We populate the taxonomy built in Section 4.3 by leveraging the links between leaf heads and clusters of categories, as well as between categories and pages. In this way, we are able to connect a given page to its proper head through an *instance-of* relation. Once the leaf type is assigned, its supertypes can be automatically inferred on account of the taxonomy hierarchy. We informally report below the foreseen procedure, which is applied to each leaf head in the set produced after stage 2 (cf. Section 4.2).

1. Extract the set of categories clustered under the given head h
2. Extract the pages linked to each category in the cluster

Table 2. Type coverage of Wikipedia articles

DBpedia	DBTax	YAGO
.513	**.963**	.673

3. For each page p:
   (a) If it is an actual page (i.e., its namespace equals to 0), then produce an assertion in the form of a triple $(p, \textit{instance-of}, h)$
   (b) If it is a category (i.e., its namespace equals to 14), recursively repeat from point 2 until the condition in point 3(a) is satisfied

5 Evaluation

The outcomes of DBTax are three-fold, namely:

- The taxonomy (T-Box) automatically generated as per stage 3 (Section 4.3) is composed of 1,902 classes
- 10,729,507 *instance-of* assertions (A-Box) are produced as output of the pages type inference step (Section 4.4). They are serialized into triples, according to the RDF data model.[13] We use the Turtle[14] syntax, which supports International Resource Identifiers (IRIs), thus fitting well for multilingual Wikipedia pages. An example is reported as follows.

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

- A total of 4,128,395 unique resources are assigned a type, 2,325,506 of which do not a have a type in the DBpedia 3.9 release

All the datasets are publicly available[15] and the source code can be checked out on GitHub.[16]

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* and *disambiguation* pages. We report the values in Table 2. DBTax clearly outperforms both DBpedia and YAGO.

5.1 T-Box Evaluation

We compare our results against the DBpedia and YAGO class hierarchies, as well as the Wikipedia category system, 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.

[13] http://www.w3.org/TR/rdf11-concepts/
[14] http://www.w3.org/TR/turtle/
[15] http://it.dbpedia.org/downloads/dbtax
[16] https://github.com/dbpedia/dbpedia-links/tree/ekaw2014/WikipediaCategoryProcessor

Table 3. T-Box evaluation results

	Class	Breakable	Valid	Specific	Broad
Wikipedia	.82	.19	.65	.66	.69
YAGO	.91	.20	.78	.40	.91
DBpedia	.95	.74	.88	.97	.81
DBTax	.91	.93	.80	.96	.35

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 to more than one classes?"* (Breakable). For the hierarchy path evaluation, we pick a random sample of 50 *leaf* classes from each ontology and generate the hierarchy path up to the root node. 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). In total, 10 evaluators participated and each question was evaluated twice. The namespaces were hidden to avoid bias and the questions were globally randomized.

Discussion. Table 3 shows the overall results. Out of the four resources, DBpedia 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 in the *Class* and *Valid* questions. DBTax provides better non-breakable classes, as it solely consists of prominent nodes and does not create too specific hierarchies (cf. the *Specific* column), as opposed to YAGO. Finally, DBTax stands last when it comes to broad hierarchies. This is due to the pruning step (cf. Section 4.3), where many nodes were removed and leaf nodes got attached to the root one.

5.2 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,[17] which serves as a bridge to a plethora of crowdsourcing channels. In this way, we are able to simultaneously determine (a) the precision of the DBTax assertions, and (b) the intuitiveness of the underlying semantics.

Task Anatomy. We randomly isolate 500 assertions from those that do not have a type counterpart in the DBpedia ontology. Hence, we consider our evaluation

[17] `http://www.crowdflower.com`

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 simplicity and atomicity. Workers are shown (1) a link to a Wikipedia page (i.e., the subject of the assertion), labeled with the word *this* in the question *"What is this?"*, and (2) a type (i.e., the object of the assertion, 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 assertion, we elicit 5 judgments, thus gathering a total of 2,500 and enabling the inter-rater agreement computation. 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 assertions. 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 flagged as untrusted and his or her judgments will be automatically discarded. In order to make our results comparable to YAGO, we conduct an identical experiment with the YAGO instance types dataset.[18] Input data, full results, interface code and screenshots are publicly available.[19]

Discussion. First, we notice that YAGO lacks of type information for 63 out of 500 pages, while our approach is always able to guess a type. Table 4 displays the results we collected. To our surprise, DBTax remarkably outperforms YAGO in terms of precision. Furthermore, we consider that recall is the fraction of pages for which the type is given. Although DBTax reaches full recall, its precision remains satisfactorily high, thus achieving an optimal trade-off. Inter-rater agreement for each question is included in the results by CrowdFlower, and we calculate the average among the whole evaluation set. Given similar agreement values, 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 Wikipedia pages are identical in both 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, even for common worldwide end users.

6 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

[18] `http://downloads.dbpedia.org/3.9/links/yago_types.ttl.bz2`
[19] `http://it.dbpedia.org/downloads/dbtax/evaluation`

Table 4. Comparative A-Box evaluation on 500 randomly selected assertions with no type coverage in DBpedia. ♠ indicates statistically significant difference with $p < 0.0005$ using χ^2 test

Resource	Precision	Agreement	Untrusted
DBTax	**.744**♠	.867	518
YAGO	.461	.871	1,358

provide 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.

Table 5. Overview of Wikipedia-powered knowledge bases. ♢ indicates a caveat.

Resource	Categories	Pages	Multilingual	3^{rd} party data
BabelNet [16]	✔	✔	✔	✔
DBpedia [12,2]	✔	✔	✔	✔
Freebase [3]	✘	✔	✔♢	✔
MENTA [13]	✔	✔	✔	✘
WiBi [8]	✔	✔	✘	✘
WikiNet [14,15]	✔	✔	✔	✘
WikiTaxonomy [22,23]	✔	✘	✘	✘
YAGO [26,11]	✔	✔	✘	✔

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 5 reports an alphabetically ordered summary of the most influential examples, which all attempt to extract structured data from Wikipedia, although with different aims. BabelNet [16] is a multilingual lexico-semantic network, which recently moved towards a Linked Data compliant representation [6]. It provides wide-coverage lexicographic knowledge in 50 languages, where common concepts and real-world entities are linked together via semantic relations. Under this perspective, BabelNet emanates from the lexical databases community, with WordNet [7] being the most mature approach. In contrast to our work, priority is given to fine-grained horizontal knowledge modeling, rather than intuitive exploration capabilities. DBpedia [12,2] 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 5), since information in other languages only appears when a Wikipedia page has an English counterpart. MENTA [13] 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. WiBi [8] attempts to build a merged 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 [14,15] 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 kind of relations. While it seems to attain wide coverage, a comparative evaluation performed in [8] highlights very low precision.

The approach that most influenced our work is YAGO [11,26]. 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. While the authors claim to have manually filtered both administrative and relational categories due to their slight presence, as an alternative we adopt a technique based on usage, detailed in Section 4.2. Similarly to us, linguistic-based processing is applied to distinguish conceptual categories from relational and topical ones.

On the other hand, the recently proposed automatic methods for type inference [18,1,19,17] may be considered as bootstrapping resources to cleanse the current state of the DBpedia ontology, at least for the classes hierarchy. Moreover, they can be used as an assisted tool to prevent redundancy, namely to alert a human contributor when he or she is trying to add some new class or property that already exists or has similar name. Hence, these efforts represent alternative solutions compared to our work.

7 Conclusion

DBTax is the outcome of a completely data-driven approach to convert the chaotic Wikipedia category system to a consistent general purpose ontology. As a result of our four-step processing pipeline, we generated a hierarchy of 1,902 classes and automatically assigned types to roughly 4 million DBpedia resources. Thus, we provide a huge coverage leap, as opposed to the current DBpedia ontology, covering only 2.2 million resources. Moreover, online evaluations in a crowdsourcing environment demonstrate that DBTax does not only outperform the manually curated DBpedia ontology and YAGO in terms of coverage and taxonomical structure, but is also outstandingly intuitive for common end users.

We envision DBTax to serve as a balance between DBpedia and YAGO, as we argue that DBpedia is very limiting and YAGO far too large for real-world use cases. For future work, we plan to merge DBTax into the DBpedia mappings wiki[20] 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 ontology pruning step. Finally, we expect to further 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 other DBpedia language chapters.

Acknowledgments We would like to thank Google for sponsoring part of this work through the Google Summer of Code 2013,[21] as well as Kasun Perera, the selected student. This work was also supported by grants from the European Union's 7th Framework Programme via the GeoKnow project (GA no. 318159).

References

1. Aprosio, A.P., Giuliano, C., Lavelli, A.: Towards an automatic creation of localized versions of dbpedia. In: The Semantic Web–ISWC 2013, pp. 494–509. Springer (2013)
2. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: 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. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. pp. 1247–1250. ACM (2008)
4. Calzolari, N., Pecchia, L., Zampolli, A.: Working on the italian machine dictionary: a semantic approach. In: Proceedings of the 5th conference on Computational linguistics-Volume 2. pp. 49–52. Association for Computational Linguistics (1973)
5. Ding, L., Lebo, T., Erickson, J.S., DiFranzo, D., Williams, G.T., Li, X., Michaelis, J., Graves, A., Zheng, J.G., Shangguan, Z., 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)
6. Ehrmann, M., Cecconi, F., Vannella, D., McCrae, J., Cimiano, P., Navigli, R.: Representing multilingual data as linked data: the case of babelnet 2.0. In: Proc. of LREC (2014)
7. Fellbaum, C.: Wordnet: An electronic lexical database. MIT Press Cambridge (1998)
8. Flati, T., Vannella, D., Pasini, T., Navigli, R.: Two is bigger (and better) than one: the wikipedia bitaxonomy project. In: Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (2014)
9. Gray, J., Chambers, L., Bounegru, L.: The data journalism handbook. " O'Reilly Media, Inc." (2012)

[20] `http://mappings.dbpedia.org`

[21] `http://www.google-melange.com/gsoc/org2/google/gsoc2013/dbpediaspotlight`

10. Haslhofer, B., Momeni, E., Gay, M., Simon, R.: Augmenting europeana content with linked data resources. In: Proceedings of the 6th International Conference on Semantic Systems. p. 40. ACM (2010)
11. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: Yago2: a spatially and temporally enhanced knowledge base from wikipedia. Artificial Intelligence 194, 28–61 (2013)
12. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., et al.: Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web (2014)
13. de Melo, G., Weikum, G.: Menta: inducing multilingual taxonomies from wikipedia. In: Proceedings of the 19th ACM international conference on Information and knowledge management. pp. 1099–1108. ACM (2010)
14. Nastase, V., Strube, M.: Transforming wikipedia into a large scale multilingual concept network. Artificial Intelligence 194, 62–85 (2013)
15. Nastase, V., Strube, M., Boerschinger, B., Zirn, C., Elghafari, A.: Wikinet: A very large scale multi-lingual concept network. In: LREC. Citeseer (2010)
16. Navigli, R., Ponzetto, S.P.: Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence 193, 217–250 (2012)
17. Nuzzolese, A.G., Gangemi, A., Presutti, V., Ciancarini, P.: Type inference through the analysis of wikipedia links. In: LDOW (2012)
18. Paulheim, H., Bizer, C.: Type inference on noisy rdf data. In: The Semantic Web–ISWC 2013, pp. 510–525. Springer (2013)
19. Pohl, A.: 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. Pollard, C.: Head-driven phrase structure grammar. University of Chicago Press (1994)
21. Pollard, C., Sag, I.A.: Information-based syntax and semantics, volume 1: Fundamentals (1987)
22. Ponzetto, S.P., Strube, M.: Deriving a large scale taxonomy from wikipedia. In: AAAI. vol. 7, pp. 1440–1445 (2007)
23. Ponzetto, S.P., Strube, M.: Taxonomy induction based on a collaboratively built knowledge repository. Artificial Intelligence 175(9-10), 1737–1756 (2011)
24. Sleator, D.D., Temperley, D.: Parsing english with a link grammar (1991)
25. Suchanek, F.M., Ifrim, G., Weikum, G.: 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. pp. 18–25 (2006)
26. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web. pp. 697–706. ACM (2007)
27. Zirn, C., Nastase, V., Strube, M.: Distinguishing between instances and classes in the wikipedia taxonomy. Springer (2008)