Type inference through the analysis of Wikipedia links

Andrea Giovanni Nuzzolese
ISTC-CNR, STLab
CS Dept., University of Bologna, Italy
nuzzoles@cs.unibo.it

Aldo Gangemi
ISTC-CNR, Semantic Technology Lab, Rome, Italy
aldo.gangemi@cnr.it

Valentina Presutti
ISTC-CNR, Semantic Technology Lab, Rome, Italy
valentina.presutti@cnr.it

Paolo Ciancarini
ISTC-CNR, STLab
CS Dept., University of Bologna, Italy
ciancarini@cs.unibo.it

ABSTRACT

DBpedia contains millions of untyped entities, either if we consider the native DBpedia ontology, or Yago plus WordNet. Is it possible to automatically classify those entities? Based on previous work on wikilink invariances, we wondered if wikilinks convey a knowledge rich enough for their classification. In this paper we give three contributions. Concerning the DBpedia link structure, we describe some measurements and notice both problems (e.g. the bias that could be induced by the incomplete ontological coverage of the DBpedia ontology), and potentials existing in current type coverage. Concerning classification, we present two techniques that exploit wikilinks, one based on induction from machine learning techniques, and the other on abduction. Finally, we discuss the limited results of classification, which confirmed our fears expressed in the description of general figures from the measurement. We also suggest some new possible directions to entity classification that could be taken.

1. INTRODUCTION

DBpedia is the largest RDF data set in Linked Data extracted from the largest existing multi-domain knowledge source edited by the crowds, i.e. Wikipedia. Part of DBpedia entities are explicitly typed with classes of the DBpedia Ontology (DBPO). This huge source of semantic data provides a powerful knowledge base that can be exploited as background knowledge for developing new generation of knowledge extraction and interaction tools. Nevertheless, most of DBpedia entities are still untyped, which limits its exploitation at its full potential. Based on previous work on wikilink invariances [13], we wondered if wikilinks convey a knowledge rich enough for their automatic classification. In this paper, we discuss both problems and potentials deriving from the current type coverage of DBpedia entities. Additionally, we present two automatic classification techniques that exploit wikilinks, one based on indiction from machine learning, and the other based on abduction. Both methods showed limited results in terms of performance (i.e., precision and recall), which can be explained by analyzing the general figures from our measurements on the DBpedia link structure. More specifically, (i) the mapping procedure between Wikipedia infobox templates and DBpedia ontology classes is conducted manually, as described in [9], resulting in a lack of ontological coverage of DBPO i.e. DBPO classes are insufficient for typing all DBpedia entities, which in turn results in lack of classification knowledge that can be used by automatic techniques for identifying type candidates for untyped entities, (ii) the granularity of the assigned types is not homogeneous e.g. some entities are typed with very general classes e.g. Person, while other similar entities have more specialized types e.g. Musician, (iii) most of the entities are untyped, hence making it hard to build a proper training set for inductive learning techniques, (iv) the distribution of link types ingoing to, and outgoing from, DBpedia entities varies between typed and untyped entities, which impacts on the ability of our abductive method to properly classify untyped entities.

After describing the resources that we have used for our research and discussing limitations and potentials emerging from our measurements on the general figures of the DBpedia link structure (Section 2), we present two methods for automatic classification of DBpedia entities and the results we have obtained (Section 3). In Section 4 we discuss related work and in Section 5 we conclude by discussing possible new approaches to DBpedia entity classification that could be taken, based on more solid grounds.

2. MATERIALS

DBpedia datasets describe about 18 million resources (3 million with an abstract and a label, less than 2 million typed), and include more than 107 million wikilinks.
The classification attempted in this paper is based on the assumption that wikilink relations in Wikipedia convey a rich knowledge that can be used to classify untyped entities referenced in those pages. In practice, given a certain entity

LDOW2012, April 16, 2012, Lyon, France.

Table 1: Analysis of DBpedia resources with respect to some relevant perspective for our work.

Perspective	# of resources/axioms
Number of wikilink axioms	107,892,317
Wikilink axioms having both the subject and the object typed with DBPO types	16,745,830
Resources used in wikilinks relations	15,944,381
rdf:type triples	6,173,940
Resources having a DBPO type	1,668,503
Resources typed with a DBPO type and used as subject of wikilinks	1,518,697

described in a Wikipedia page, our classification grounds on the analysis of incoming and outgoing wikilinks from and to that certain page.

For this analysis we used DBpedia [9], the RDF [10] Linked Data [1] that contains structured information extracted from Wikipedia.

The types used to classify DBpedia resources are the classes of the DBpedia ontology [1] (DBPO). DBPO ontology is represented in OWL [11] and covers 272 classes.

The `rdf:type` statements from DBpedia resources to DBPO classes (available in the dataset `dbpedia_instance_types_en`) are the result of hand-generated mappings of Wikipedia infoboxes to DBPO, and have been generated for 1,668,503 DBpedia resources[2].

Those statements cover only a subset of the 15,944,381 DBpedia resources available in the `dbpedia_page_links_en` dataset. Hence, only 15.52% of resources in the DBpedia data set of wikilinks are typed with DBPO classes. In the work described in this paper we investigate how to assign a DBPO type to the remaining 84.48% untyped DBpedia resources.[3].

Table 2 shows further details about how resources are organized in DBpedia. It emerges that the number of wikilinks is much bigger (107,892,317) than the number of `rdf:type` triples (6,173,940). When we take into account only DBPO type we can observe how the number of Wikilink axioms having both the subject and the object typed with DBPO types is limited to 16,745,830.

Furthermore, we analyzed the distribution of wikilinks across typed/untyped resources. The average percentage of fully DBPO typed wikilinks (i.e., wikilink axioms having both the subject and the object typed with a DBPO class) per resource is 66% with respect to the total number of wikilink axioms per resource. Instead, the average percentage of wikilink axioms outgoing from untyped to typed resources is 37% per resource with respect to the total number of wikilink axioms per resource. This means that the ratio between typed and untyped outgoing links is 23 for typed resources vs. 13 for untyped ones.

The reason of this unbalance can be hypothesized to be caused by the high frequency of *homotypes*, i.e. wikilinks that have the same type on both the subject and the object of the triple. If this hypothesis is confirmed, untyped resources should have a high ratio of untyped outgoing links. As a matter of fact, homotypes are actually very frequent (usually the most frequent, or in the top 3) wikilink types (this observation has been made in the research reported by [13]). Therefore, such distribution of wikilinks for typed and untyped resources is unbalanced.

This means that if we use wikilinks and types as the features for training or designing a good inductive model based on the corpus of typed resources, a bias is created on the appropriateness of such a model for classifying untyped resources. However, we wanted to check anyway: i) what precision/recall can be obtained when using wikilink structures as features for creating a type induction model from typed DBpedia resources, even considering the bias constituted by the 34% untyped wikilinks; and ii) how much the larger bias (63%), constituted by untyped wikilinks on untyped resources, would affect the precision/recall established on typed resources.

A part of our wikilink analysis for DBpedia entity classification made use of 187 Knowledge Patterns that have been extracted from DBpedia wikilink datasets, which are called Encyclopedic Knowledge Patterns (EKPs) [4] [13]. EKPs allow to fetch most relevant entity types that provide an effective and intuitive description of entities of a certain type. As discussed in the next section, EKPs provide a background knowledge to our method of abductive classification.

3. METHODS AND RESULTS

The classification of DBpedia entities relies on an ontology mapping task which defines how Wikipedia infobox templates are mapped to classes of the DBpedia ontology. These mappings are manually specified using the DBpedia Mapping Language [5]. The mapping language makes use of MediaWiki templates that allow to map infoboxes to DBpedia ontology classes. The mappings cover only a small subset of all Wikipedia infoboxes. As a result, so far, only a small subset of all DBpedia entities (1,668,503 of 15,944,381) is typed with a class of the DBpedia ontology. Probably the effort spent in manually writing mappings for the classification of DBpedia entities with respect to the DBpedia ontology is too expensive and the granularity and the appropriateness of obtained typings are not exhaustive. As an example, `dbpedia:Walt_Disney` [6] is typed as `dbpo:Person`, which is doubtlessly correct, but also trivial and less appropriate than the existing `dbpo:ComicsCreator type`.

Our work is based on the intuition that wikilink relations in DBpedia, i.e. instances of the `dbpo:wikiPageWikiLink` property, convey a rich knowledge that can be used for classifying DBpedia entities, but at the same time it is very difficult to find a good training set by using only typed wikilinks. A first reason is that typed resources having wikilinks are only the 15.52% of the total resources used in the wikilink data set. A second one derives from the fact untyped resources, i.e., the resources we are interested to type, have

[1] http://wiki.dbpedia.org/Ontology

[2] such figure holds for the typed resources in the English version of DBpedia 3.6

[3] Actually, excluding some entities that are not relevant: images, categories, and disambiguation pages

[4] The EKP resource is available as a set of OWL ontologies at http://ontologydesignpatterns.org/ekp/owl/

[5] http://mappings.dbpedia.org/index.php/Main_Page

[6] The prefixes `dbpo` and `dbpedia` stand for `http://dbpedia.org/ontology/` and `http://dbpedia.org/resource/` respectively.

Table 2: Number of individuals chosen for generating the training examples relatively to the classes Place, Person, Work, Organisation and Activity.

Class	# of individuals	# of trained individuals
Place	525,786	105,157
Person	416,079	83,215
Work	262,662	52,532
Organisation	169,338	33,867
Activity	1,380	276

only one third of their total wikilinks typed. In order to test this intuition, we have investigated and compared type induction methods over DBpedia entities based on two inference types:

- Inductive classification: works moving from specific observations to broader generalizations and theories, and can be informally defined as a "bottom-up" approach;
- Abductive classification: works from more general rules (assumed from previous cases), in order to infer presumably specific facts. In this case we have used EKPs and homotypes as background knowledge.

Considering that DBPO has 272 classes, and that automatic classification on 272 classes is very difficult, we have focused mainly on classification of entities with respect to the DBPO top-level. The granularity of the classification is solved in this case by adopting a hierarchical-iterative type induction derived by the class hierarchy in DBPO. This means that the classification starts from the top level of the DBpedia ontology composed by 27 classes, and then iteratively goes on trying to classify an entity with one of the sub-classes of the class selected in the previous iteration.

3.1 Inductive type inference

Hybridization of Machine Learning (ML) techniques with the Semantic Web is quite effective [5], therefore we started our investigation trying to use a ML-based approach to the classification of DBpedia entities. We have used the k-Nearest Neighbor (NN) algorithm for classifying DBpedia entities based on the closest training examples in the labeled feature space, and by assigning the most voted class among the training examples.

We have designed two inductive classification experiments based on the NN algorithm: (i) a baseline experiment configured to classify DBpedia individuals based on 272 features, i.e., all the DBPO classes. (ii) An experiment based on the top-level classes in the DBPO taxonomy, i.e., 27 classes, aimed to simplify the classification with less features to investigate. In both cases the training sample has been built with the same approach described as follows: (i) the 20% of the individuals of each class has been used for populating the training sample. Table 3.1 shows how many individuals have been chosen with respect to the classes Place, Person, Work, Organisation and Activity. (ii) The NN algorithm has been trained on a labeled feature space model in which the individuals of the training sample and the classes of the DBPO were the rows and the feature columns of the model respectively. For each individual we labeled the corresponding row with its known DBPO class and we then analyzed its graph of wikilinks in order to fill the matrix resulting from the space-vector model. This was done marking with either 0 or 1 each intersection cell between a row corresponding to an individual and a feature corresponding to a DBPO class with the following criteria:

Table 3: Some of the wikilinks outgoing from `dbpedia:Steve_Jobs` and their associated type.

Links	Class
...	...
`dbpedia:Apple_Inc.`	`dbpo:Company`
`dbpedia:NeXT`	`dbpo:Company`
`Cupertino,_California`	`dbpo:City`
`Forbes`	`dbpo:Magazine`
...	...

- 0 means that no wikilink exists between the selected individual and any other individual typed with the corresponding DBPO class used as feature;
- 1 means that at least one wikilink exists between the selected individual and any other individual typed with the corresponding DBPO class used as a feature.

As an example, we may want to built the feature model of the wikilinks of the entity `dbpedia:Steve_Jobs` with respect to the classes `dbpo:Mammal`, `dbpo:Scientist`, `dbpo:Company`, `dbpo:Drug`, `dbpo:City`, `dbpo:Magazine`. We fetch all the types related to the outgoing wikilinks from `dbpedia:Steve_Jobs` with a simple SPARQL query like the following:

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
SELECT ?link ?type WHERE {
    GRAPH<dbpedia_page_links_en> {
        dbpedia:Steve_Jobs ?prop ?link
    } .
    GRAPH<dbpedia_instance_types_en> {
        ?link a ?type
    }
}
```

Supposing that all the wikilinks retrieved (actually, those shown are only a subset) are the ones showed in table 3.1 the resulting row in feature space model deriving from `dbpedia:Steve_Jobs` will be the following:

```
             Mammal  Scientist  Company  Drug  City  Magazine
 Steve_Jobs    0         0         1      0     1       1       Person
```

After the training, for each class the classification function generates what is called a mean, i.e., the reference value used to test similarities and then to classify untyped entities. The classification has been performed by estimating the euclidean distance of the features of unlabeled individuals with respect to each mean. The lower values of euclidean distance have been chosen to classify individuals.

The performance of this approach has been evaluated on the remaining 80% untyped individuals. We remind that such evaluation is made considering the "best case" of resources with a known type, in which the percentage of typed wikilinks is high (66%). The precision deriving from the baseline experiment has been 31.65%. The precision of the classification of individuals with respect to the DBPO top-level has been 40.27%.

3.2 Abductive type inference

Abduction was introduced by Peirce [7, 14], and refers to a process oriented to an explanatory hypothesis about the precondition $\mathcal{P}$ reasoning on the consequence $\mathcal{C}$. Compared to pure deduction, induction and abduction have a lower strength, much are practical when the set of assumptions is not complete with respect to the observed world. Induction cannot be made fully conclusive, since the inference from a set of cases can be only made certain in a closed world (which is not the case with the Web). Abduction, on its turn, is formally equivalent to the logical fallacy *affirming the consequent* or *Post hoc ergo propter hoc*, because there are multiple possible explanations for $\mathcal{C}$. In other words, abducing $\mathcal{P}$ from $\mathcal{C}$ involves determining that $\mathcal{C}$ is sufficient (or nearly sufficient), but not necessary, for $\mathcal{P}$.
We have used an abductive approach to infer the type of DBpedia entities with two classification methods:

- EKP-based. We assumed Encyclopedic Knowledge Patterns (EKPs) extracted from wikilink relations in Wikipedia [13] as our background knowledge, and as the adbuctive consequence $\mathcal{C}$, on which we infer the type of entities. In this context, entity types are our precondition $\mathcal{P}$. We want to infer types by analysing the similarity between rules derived by EKPs, and the configurations of wikilink relations obtained from untyped entities in DBpedia.

- *Homotype*-based. We define homotypes as wikilinks that have the same type on both the subject and the object of the triple. Since homotypes are usually the most frequent (or in the top 3) wikilink types [13], we want to detect emerging homotypes for untyped resources by summing the number of outgoing and incoming wikilinks.

EKP-based type adbuction. EKPs are defined as sets of type paths that occur most often above a certain threshold t [13]. In this context, a path relates the types of two entities having a wikilink. The frequency of paths composing an EKP is calculated by the *path popularity*. The path popularity is defined as the ratio of how many distinct resources of a certain type participate as subject in a path to the total number of resources of that type. We use path popularity values of each EKP in order to estimate the similarity between the wikilinks of an entity and a EKP. For doing that we build a labeled feature space model $i \times j$ composed by:

- i rows, each one labeled by a different top-level class of the DBPO taxonomy. This means $0 \leq i \leq 27$;

- j features as columns consisting in the number of all the classes in the DBPO. This means $0 \leq j \leq 272$;

- cell values that contain path popularity values occurring between the type in the row and the type in the column.

The following is an example of feature space model built for the class labels `Person`, `Place` and `Oganisation` over the features `Event`, `Work`, `Organisation`, `Person`, `Activity`, `Place`:

	Event	Work	Organisation	Person	Activity	Place
...	...	...	...	...	...	...
Person	4.45	8.4	22.2	18.29	6.8	27.5
Organisation	3.21	7.78	24.13	13.23	2.5	31.93
Place	3.14	1.86	8.71	8.74	1.1	61.15
...	...	...	...	...	...	...

In the abductive approach, differently from the inductive one, the labeled feature space model is not used for training the classification function. In fact, it already provides general rules based on path popularity values over the wikilink domain used to infer types.
Instead, entities to be classified are represented in the same way as in the inductive approach. Therefore, they are vector models of length j, where j is the number of features used in the model. Values in vectors consists of either 0 or 1, in which, again:

- 0 means that no wikilink exists between the selected individual and any other individual typed with the corresponding DBPO class used in the feature;

- 1 means that at least one wikilink exists between the selected individual and any other individual typed with the corresponding DBPO class used in the feature.

Individuals are classified by finding similarities to EKPs by applying a similarity metric based on the analysis of path popularity values. This suggests what is the closest EKP to the configuration of wikilinks that the individual presents. Given a labeled feature space model M and a vector I defined as described before, the similarity function is defined as follows:

$$S(i) = \frac{\sum_{j=0}^{F} M_{i,j} * I_j}{\sum_{j=0}^{F} M_{i,j}}$$

where

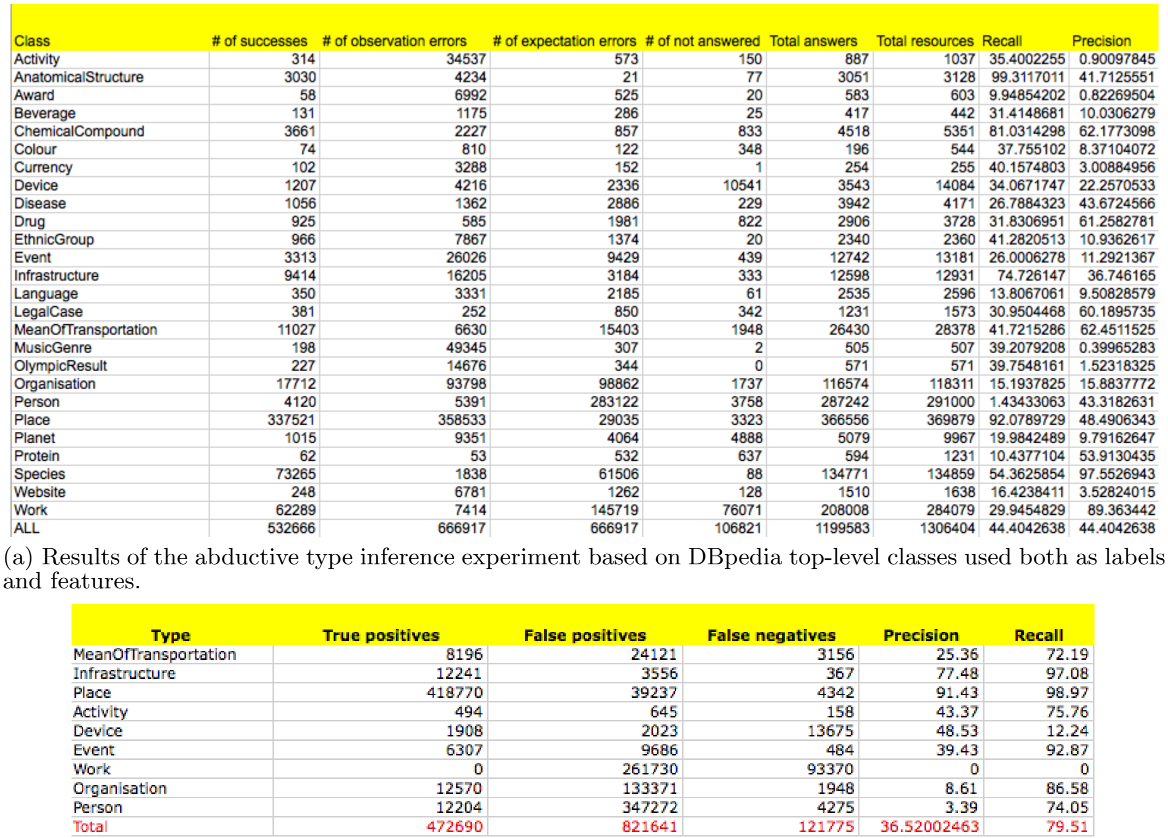
- F is the number of DBPO classes used as features;

- $M_{i,j}$ is the path popularity value between the i-th class used as label and the j-th class used as feature;

- I_j is the 0 or 1 value that corresponds to the fact that the entity has or has not a wikilink relation with an entity typed with the j-th class used as feature;

- F is the number of features used in the model and $0 \leq j \leq F$;

- $0 \leq i \leq L$, where L is the number of available labels in the model.



Class	# of successes	# of observation errors	# of expectation errors	# of not answered	Total answers	Total resources	Recall	Precision
Activity	314	34537	573	150	887	1037	35.4002255	0.90097845
AnatomicalStructure	3030	4234	21	77	3051	3128	99.3117011	41.7125551
Award	58	6992	525	20	583	603	9.94854202	0.82269504
Beverage	131	1175	286	25	417	442	31.4148681	10.0306279
ChemicalCompound	3661	2227	857	833	4518	5351	81.0314298	62.1773098
Colour	74	810	122	348	196	544	37.755102	8.37104072
Currency	102	3288	152	1	254	255	40.1574803	3.00884956
Device	1207	4216	2336	10541	3543	14084	34.0671747	22.2570533
Disease	1056	1362	2886	229	3942	4171	26.7884323	43.6724566
Drug	925	585	1981	822	2906	3728	31.8306951	61.2582781
EthnicGroup	966	7867	1374	20	2340	2360	41.2820513	10.9362617
Event	3313	26026	9429	439	12742	13181	26.0006278	11.2921367
Infrastructure	9414	16205	3184	333	12598	12931	74.726147	36.746165
Language	350	3331	2185	61	2535	2596	13.8067061	9.50828579
LegalCase	381	252	850	342	1231	1573	30.9504468	60.1895735
MeanOfTransportation	11027	6630	15403	1948	26430	28378	41.7215286	62.4511525
MusicGenre	198	49345	307	2	505	507	39.2079208	0.39965283
OlympicResult	227	14676	344	0	571	571	39.7548161	1.52318325
Organisation	17712	93798	98862	1737	116574	118311	15.1937825	15.8837772
Person	4120	5391	283122	3758	287242	291000	1.43433063	43.3182631
Place	337521	358533	29035	3323	366556	369879	92.0789729	48.4906343
Planet	1015	9351	4064	4888	5079	9967	19.9842489	9.79162647
Protein	62	53	532	637	594	1231	10.4377104	53.9130435
Species	73265	1838	61506	88	134771	134859	54.3625854	97.5526943
Website	248	6781	1262	128	1510	1638	16.4238411	3.52824015
Work	62289	7414	145719	76071	208008	284079	29.9454829	89.363442
ALL	532666	666917	666917	106821	1199583	1306404	44.4042638	44.4042638

(a) Results of the abductive type inference experiment based on DBpedia top-level classes used both as labels and features.

Type	True positives	False positives	False negatives	Precision	Recall
MeanOfTransportation	8196	24121	3156	25.36	72.19
Infrastructure	12241	3556	367	77.48	97.08
Place	418770	39237	4342	91.43	98.97
Activity	494	645	158	43.37	75.76
Device	1908	2023	13675	48.53	12.24
Event	6307	9686	484	39.43	92.87
Work	0	261730	93370	0	0
Organisation	12570	133371	1948	8.61	86.58
Person	12204	347272	4275	3.39	74.05
Total	472690	821641	121775	36.52002463	79.51

(b) Results of the abductive type inference experiment based on classes `Activity`, `Device`, `Event`, `Infrastructure`, `MeanOfTransportation`, `Organisation`, `Person`, `Place` and `Work` used both as labels and features.

Figure 1: Results of the experiments based on abductive type inference.

The function S returns a similarity score calculated locally with respect to a type, while we are interested in the score that maximizes the similarity, hence:

$$T = \max S(i), \forall\ 0 \leq i \leq L$$

Assuming that for a certain entity Y with respect to the types `Event`, `Work`, `Organisation`, `Person`, `Activity`, `Place` has the following wikilink configuration:

	Event	Work	Organisation	Person	Activity	Place
Y	0	1	0	1	0	0

We obtain that the entity Y is classified as Person, since the value $S(Person) = (1 * 8.4 + 1 * 18.29)/87.64 = 0.3$ is the highest similarity value.

In our first abductive experiment we have focused only on the top-level classes of the DBpedia taxonomy both for class labels and for features. Both L and F are then equal to the number of the top-level classes in the DBpedia taxonomy, i.e., 27.

The precision and recall of this experiment are both, as shown in figure 1(a), 44.4%. The figure shows, besides the precision and recall relative to single classes, additional metrics, like the number of true positives, false positives, false negatives, etc.

What also emerges from figure 1(a) is that a small subset (`Device`, `Event`, `Infrastructure`, `MeanOfTransportation`, `Organisation`, `Person`, `Place` and `Work`) of classes contains the majority of the individuals. For that reason, we have tried to apply the abductive approach only to that subset plus the class `Activity`, which instead has a very low precision.
In this second experiment we have used those 9 classes as labels for classifying DBpedia entities over the same 9 classes as features. Under these assumptions, the global precision has fallen from 44.4% of the previous experiment down to 36.5, while the recall has grown from 44.4% up to 79.5%. Figure 1(b) shows the results of this experiment.

Homotype-based abduction. In this case we use abduction in order to guess the type of DBpedia individuals by using homotypes as background knowledge. An homotype is usually the most frequent (or in the top 3) wikilink type. For that reason, we want to infer the type of a resource by detecting the emerging homotype. If most of the incoming and outgoing wikilinks of a resource R is of a same type T, then we can infer, under the homotype assumption, that the

type of R is T.
Given an individual i, we can then define the set W of all the classes used to type individuals having an incoming or an outgoing wikilink relation with i, as:

$$W(i) = \{\langle n, X\rangle | n = \#x \in X \text{ s.t } i \rightarrow x \vee i \leftarrow x\}$$

where

- the notation $\in$ stands here for the `rdf:type` property;
- the $\rightarrow$ and $\leftarrow$ stand for outgoing and incoming `dbpo:wikiPageWikiLink` properties;
- n is the number of wikilinks typed by a same class X;
- $< n, X >$ is any distinct couple that states how many instances of the class X occur in the wikilinks of i.

The homotype range H, emerging for an individual i, is formalized as the type X having the highest value n in $W(i)$, i.e.,

$$\begin{aligned} H(i) = \{X \quad | \quad & \langle n, X\rangle \in W(i) \\ & \implies \forall \langle n', X'\rangle \in W(i)\ , n \geq n'\} \end{aligned}$$

where, the notation $\in$ has here the classical meaning from set theory. Figure 2 shows how the homotype range is selected grouping the sum of outgoing and incoming wikilinks for the resource `dbpedia:Steve_Jobs` grouped by their types. According to the definition of homotype given, the resource `dbpedia:Steve_Jobs` as shown in figure 2 should be typed with the class `dbpo:Person`.

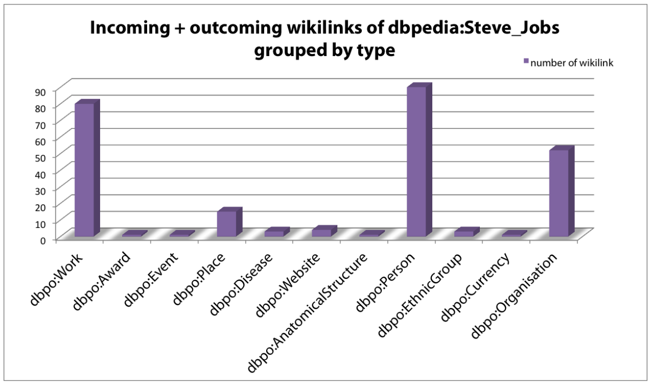
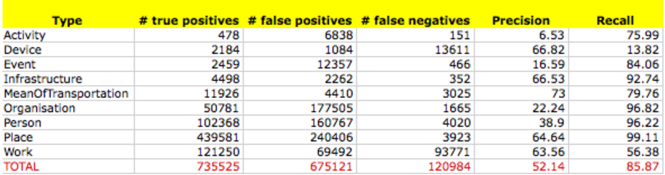


Figure 2: Incouming + outgoing wikilinks of `dbpedia:Steve_Jobs` grouped by their types.

For the classification experiment based on the homotype definition, we have defined a threshold ϵ that represents the minimum number of wikilinks that a resource should have in order to be classifiable. In fact, below a certain number of wikilinks, the homotype is less distinctive. The threshold has been fixed to be $\epsilon\ = 10$ because, as reported by [13], the average number of outgoing wikilinks per resource in the `dbpedia_page_links_en` data set of DBpedia is 10.
The homotype-based classification of individuals from the control set (with a known type) from the `dbpedia_instance_type_en` data set produced a global precision of 52.14% and a global recall of 85.87%. Figure 3 shows the details regarding global and local precision, recall and also reports the number of true positives, false positives and false negatives.
The homotype-based approach produces a side-effect. In fact, more than one class can emerge with the same frequency of wikilinks in H. In some cases, we get ex-aequo classifications, i.e. multi-typings. Multi-typing introduces noise that produces a higher number of false positives. For example, if an entity e, whose actual type is T_1, is classified with types T_1, T_2 and T_3, then T_1 will be counted as a true positive, but at the same time T_2 and T_3 will be counted as false negatives. In general multi-typings are not desirable. Figure 4 shows the number of ex-aequo typings for each cluster found, i.e. 88,845 occurrences with 2 ex-aequo classes, 11,572 occurrences with 3 ex-aequo classes, 1,118 occurrences with 4 ex-aequo classes, 44 occurrences with 5 ex-aequo classes and, finally, 1 occurrence with 6 ex-aequo classes, for a total of 101,580 ex-aequo classifications.



Type	# true positives	# false positives	# false negatives	Precision	Recall
Activity	478	6838	151	6.53	75.99
Device	2184	1084	13611	66.82	13.82
Event	2459	12357	466	16.59	84.06
Infrastructure	4498	2262	352	66.53	92.74
MeanOfTransportation	11926	4410	3025	73	79.76
Organisation	50781	177505	1665	22.24	96.82
Person	102368	160767	4020	38.9	96.22
Place	439581	240406	3923	64.64	99.11
Work	121250	69492	93771	63.56	56.38
TOTAL	735525	675121	120984	52.14	85.87

Figure 3: Precision and recall of the homotype-based classification.

In order to reduce the noise of multi-typings in the evaluation of the homotype-based approach, we have adjusted the performance analysis by applying the following criteria in case of ex-aequo:

- if among the ex-aequo classes there is the actual type of the entity, then count 1 true positive and 0 false positives;
- if among the ex-aequo classes there is not the actual type of the entity, then count 0 true positives and 1 false positive;

This increased the precision up to 55.07%.

Figure 5 shows the trend of the precision and recall through the various experiments. Considering the abductive approach has reported the best results, we have decided to run both the EKP-based and homotype-based classification on a sample of 1,000 untyped resources, to be manually evaluated. The classifier was again limited to the 9 core top-level classes as described before.
Manual evaluation reported precision and recall with EKP-based classification as 5.98% and 7.9% respectively, while with homotype-based classification as 13.93% and 65.51% respectively. Figure 6 shows the details of the results of the classifications performed on the sample of 1,000 untyped resources for the 9 top-level classes.

4. RELATED WORK

There is valuable research on knowledge extraction based on the exploitation of Wikipedia. Ontology mining aims at discovering hidden knowledge from ontological knowledge bases

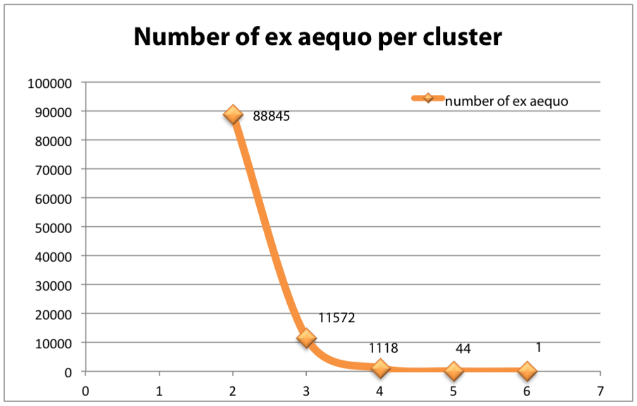


Figure 4: Distribution of ax aequo classification among the 5 clusters found.

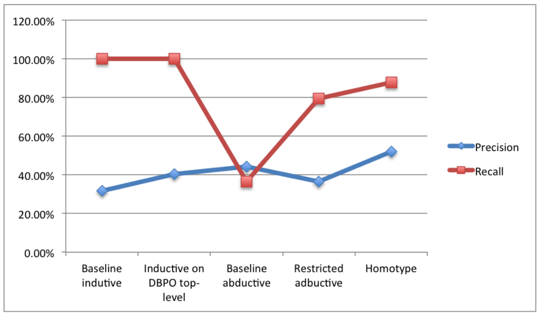
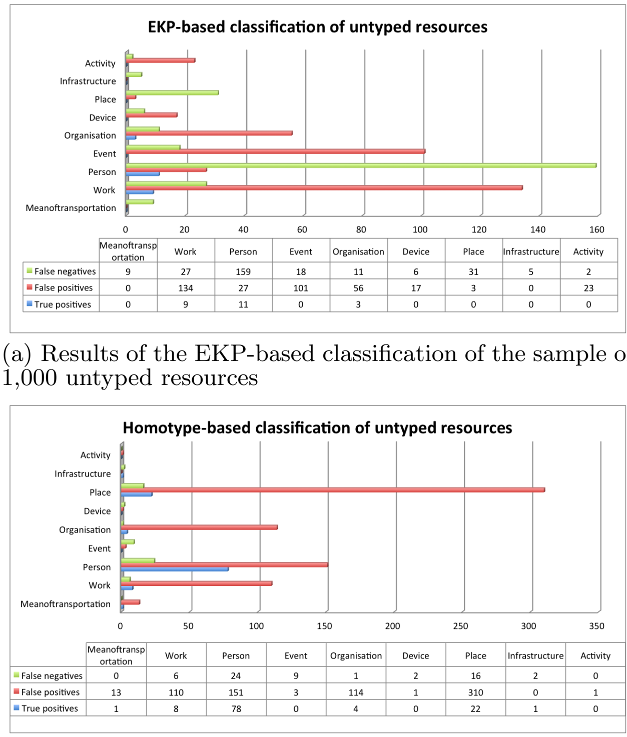


Figure 5: Precision and recall trend through the various experiments.



	Meanoftransportation	Work	Person	Event	Organisation	Device	Place	Infrastructure	Activity
🟩 False negatives	9	27	159	18	11	6	31	5	2
🟥 False positives	0	134	27	101	56	17	3	0	23
🟦 True positives	0	9	11	0	3	0	0	0	0

(a) Results of the EKP-based classification of the sample of 1,000 untyped resources

	Meanoftransportation	Work	Person	Event	Organisation	Device	Place	Infrastructure	Activity
🟩 False negatives	0	6	24	9	1	2	16	2	0
🟥 False positives	13	110	151	3	114	1	310	0	1
🟦 True positives	1	8	78	0	4	0	22	1	0

(b) Results of the Homotype-based classification of the sample of 1,000 untyped resources

Figure 6: Results of the classifications over the sample of 1,000 DBpedia untyped resources with the EKP-based classifier 6(a) and homotype-based one 6(b)

by using data mining and machine learning techniques [5]. [4] proposes an extension of the *k-Nearest Neighbor* algorithm for Description Logic KBs based on the exploitation of an *entropy*-based dissimilarity measure, while [2, 6] makes use of Support Vector Machine (SVN) [3] rather than NN to perform automatic classification. SVM performs instance classification by implicitly mapping (through a kernel function) training data to the input values in a higher dimensional feature space where instances can be classified by means of a linear classifier. The main difference between these and our approach is that they analyse all semantic properties used for linking individuals, while we have look for invariances on the usage of only one property, i.e., `dbpo:wikiPageWikiLink`, which flattens the reasoning space. The current procedure for assigning type to DBpedia entity is completely manual. As extensively described in [9], a limited number of infobox templates have been defined based on empirical observation of invariances in infoboxes of Wikipedia pages having the same ontological type. Based on previous work on wikilink invariances [13], we investigate the automatic classification of DBpedia entities. The ontology that we use for the classification is the DBpedia Ontology (DBPO) that results from the manual procedure described above, hence we inherit its limited ontological coverage. [16] presents YAGO, an ontology extracted from Wikipedia categories and infoboxes that has been combined with taxonomic relations from WordNet. Here the approach can be described as a reengineering task for transforming a thesaurus, i.e. Wikipedia category taxonomy, to an ontology, which required accurate ontological analysis. There is a significant overlap between YAGO and DBPO typed entities, and entities that have only YAGO classes cover a small part of the entities untyped with DBPO. Furthermore, there is a lack of mapping between YAGO and DBPO, which makes it difficult to exploit their merged coverage. Yago has a larger coverage than DBPO, but it has only an overlap with DBPO coverage; moreover, the granularity of Yago categories is finer, and not easily reusable, because the top level is very large. The size of the ontology, and the fact that Yago adopts multi-typing, complicate helplessly the task of automatic classification of types.

5. DISCUSSION AND CONCLUSION

We want to discover the types of all DBpedia resources. Currently only a subset of them (about 15%, about 22% in recent version 3.7) have explicit types that are coherently organized into an ontology (DBpedia Ontology, DBPO).

We investigated type inference in two directions: (i) an inductive approach typical of machine learning; (ii) an abductive approach, in which we firstly used two already available feature sets over the wikilink domain in order to infer types: (a) the EKPs [13] that have been extracted from Wikipedia with a statistical analysis over type paths generated by wikilinks, and (b) the notion of *emerging homotype*, i.e. a link that has the same types in the subject and object of the RDF triple.

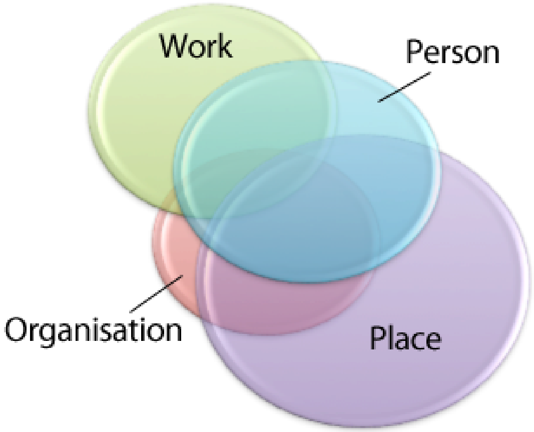


Figure 7: Overlaps of classified entities among the classes `Place`, `Person`, `Organisation` and `Work`.

We observed two possible classification biases in the DBpedia datasets. The first bias is the ratio between the number of typed resources having wikilinks and the total number of resources with wikilinks, i.e., 1,518,697 out of more than 15 million. We suspected that this bias may be the result of a partial ontological coverage, which derives from the manual mappings of Wikipedia infoboxes used to extract DBpedia resources to DBPO. The second, related, bias comes from the ratio between the average typed wikilinks owned by typed resources and the average typed wikilinks owned by untyped resources resources, that is 66% for the former and 37% for the latter.

The results seem to confirm the factor caused by those biases over the classification. In fact, while the results in classifying a test set of typed DBpedia resources – precisions 44.4% and 55.07% with EKPs and homotypes respectively – are relatively satisfactory (specially considering the amount of classes), we observed a dramatic fall of the precision in classifying untyped DBpedia resources, decreasing to 5.98% with EKPs and 13.93% with homotypes.

This means that, besides being still valid the results about the cognitive soundness of EKPs in providing an effective and intuitive description of entities of a certain type (as reported in [13]), our hypothesis about the distinctive capacity of EKPs is weak. This seems due to wide overlaps among EKPs. The same overlaps emerge when applying homotype-based classification as well. Figure 7 shows the overlaps among the 4 largest classes of DBpedia, i.e., `dbpo:Place`, `dbpo:Person`, `dbpo:Organisation` and `dbpo:Work`.

Notice that the decrease in precision from the test set to the untyped resource set spans between -38% and -41% across the different approaches, probably revealing an approximate 40% DBpedia resources that are *true negatives* with reference to the existing DBPO, so providing a rough quantification of the missing ontological coverage of the current DBPO.

For instance, a quick exploration of untyped resources immediately evidences the need for types representing important notions such as *Plan, Agreement, ScientificDiscipline, Collection, Concept, etc.*. It's quite interesting to remark that these notions are on one hand harder to generalize from infoboxes, but are also relatively established in existing foundational ontologies and ontology design patterns [8].

An area of improvement for DBPO and DBpedia typing is therefore an extension of the ontology and the ways to use the extension to type untyped resources.

A second area of improvement might be related to the imprecision deriving from the massive overlap among the abovementioned four core classes of DBPO.

However, we might consider a more critical attitude. While we deem important the role of ontologies in accurately distinguishing semantic types of things, specially in domains and tasks requiring fine-grained types, it might be interesting to explore an alternative vision of *systematically ambiguous* ontologies, where some types tend to merge because of their mutual dependence in the real world. For example, an organization is typically dependent on persons, places, and works, which on their turn can be often dependent on it: social organization is a major example.

Shifting our perspective, the distinctivity weakness of both EKPs and homotypes in classifying DBpedia resources can find some basis in recent work [15], which confirms known semiotic assumptions about the centrality of (systematic) ambiguity in language, and poses significant challenges to ontological theories assuming semantic "pedantry". An interesting research topic in ontology design may investigate the consequences of assigning high value to "clusterability" of (systematically dependent) meaning dimensions rather than to their distinctivity.

There are many directions that these results open up. Some of them include the update of EKPs to DBpedia 3.7 and a further analysis in order to discover more distinctive features in their extraction. Another direction involves a mixed approach based both on inference and crowdsourcing through, for instance, exploratory search methods on Linked Data following the direction we took with Aemoo [12]. Other useful directions include use of indexing techniques, deep parsing of natural language, or social network analyses.

6. REFERENCES

[1] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data-The story so far. *International Journal on Semantic Web and Information Systems*, 4(2):1–22, 2009.

[2] S. Bloehdorn and Y. Sure. Kernel Methods for Mining Instance Data in Ontologies. In K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, G. Schreiber, and P. Cudré-Mauroux, editors, *Proceedings of the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference (ISWC 2007 + ASWC 2007)*, volume 4825 of *Lecture Notes in Computer Science*, pages 58–71, Busan, Korea, November 2007. Springer Verlag.

[3] N. Christianini and J. Shawe-Taylor. *Support Vector Machines and Other kernel-based Learning Methods.* Cambridge University Press, 2000.

[4] C. d'Amato, N. Fanizzi, and F. Esposito. Query Answering and Ontology Population: an Inductive Approach. In M. Hauswirth, M. Koubarakis, and S. Bechhofer, editors, *Proceedings of the 5th European Semantic Web Conference (ESWC 2008)*, volume 5021 of *Lecture Notes in Computer Science*, Tenerife, Spain, June 2008. Springer Verlag.

[5] C. d'Amato, N. Fanizzi, and F. Esposito. Inductive Learning for the Semantic Web: What does it buy? *Semantic Web*, 1(1):53–59, 2010.

[6] N. Fanizzi, C. d'Amato, and F. Esposito. Statistical Learning for Inductive Query Answering on OWL Ontologies. In A. P. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. W. Finin, and K. Thirunarayan, editors, *Proceedings of the 7th International Semantic Web Conference (ISWC 2008)*, volume 5318 of *Lecture Notes in Computer Science*, pages 195–212, Karlsruhe, Germany, October 2008. Springer.

[7] H. Frankfurt. Peirce's notion of abduction. *The Journal of Philosophy*, 55(14):593–597, 1958.

[8] A. Gangemi and V. Presutti. *Handbook on Ontologies*, chapter Ontology Design Patterns. Springer, 2nd edition, 2009.

[9] J. Lehmann, C. Bizer, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - A Crystallization Point for the Web of Data. *Journal of Web Semantics*, 7(3):154–165, 2009.

[10] M. Miller and M. F. RDF Primer. W3c recommendation, W3C, feb 2004. http://www.w3.org/TR/2004/REC-rdf-primer-20040210/.

[11] B. Motik, B. Parsia, and P. F. Patel-Schneider. OWL 2 web ontology language structural specification and functional-style syntax. W3C recommendation, W3C, Oct. 2009. http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/.

[12] A. Musetti, A. G. Nuzzolese, F. Draicchio, V. Presutti, E. Blomqvist, A. Gangemi, and P. Ciancarini. Aemoo: Exploratory search based on knowledge patterns over the semantic web. *Semantic Web Challenge*.

[13] A. G. Nuzzolese, A. Gangemi, V. Presutti, and P. Ciancarini. Encyclopedic Knowledge Patterns from Wikipedia Links. In L. Aroyo, N. Noy, and C. Welty, editors, *Proceedings of the 10th International Semantic Web Conference (ISWC2011)*, pages 520–536. Springer, 2011.

[14] C. Peirce. *Pragmatism as a principle and method of right thinking: The 1903 Harvard lectures on pragmatism*. State Univ of New York Pr, 1997.

[15] S. T. Piantadosi, H. Tily, and E. Gibson. The communicative function of ambiguity in language. *Cognition*, 122(3):280 – 291, 2012.

[16] F. Suchanek, G. Kasneci, and G. Weikum. Yago - A Large Ontology from Wikipedia and WordNet. *Elsevier Journal of Web Semantics*, 6(3):203–217, 2008.