# Extending DBpedia with Wikipedia List Pages

Heiko Paulheim and Simone Paolo Ponzetto

University of Mannheim, Germany
Research Group Data and Web Science
{heiko,simone}@informatik.uni-mannheim.de

**Abstract.** Thanks to its wide coverage and general-purpose ontology, DBpedia is a prominent dataset in the Linked Open Data cloud. DBpedia's content is harvested from Wikipedia's infoboxes, based on manually created mappings. In this paper, we explore the use of a promising source of knowledge for extending DBpedia, i.e., Wikipedia's list pages. We discuss how a combination of frequent pattern mining and natural language processing (NLP) methods can be leveraged in order to extend both the DBpedia ontology, as well as the instance information in DBpedia. We provide an illustrative example to show the potential impact of our approach and discuss its main challenges.

**Keywords:** Ontology Learning, Frequent Pattern Mining, Wikipedia, DBpedia

## 1 Introduction

In the last few years, DBpedia [1] has become one of the most prominent datasets in the Linked Open Data (LOD) cloud, and it has been successfully used for a variety of high-end complex intelligent tasks such as open-domain question answering [2], topic labeling [6], web search result clustering [12], and interpreting statistical data [9]. This impact is arguably due to the fact that DBpedia is able to provide vast amounts of structured knowledge, which is embedded within a manually validated ontological model. However, questions remains on whether it can be further improved, e.g., by harvesting additional information in order to provide even more coverage.

One of the key characteristics of DBpedia, for instance, is that it is extracted from Wikipedia's infoboxes, which are manually mapped to classes and predicates in the DBpedia ontology. While infoboxes provide a highly structured source of information, they just cover part of the information encoded within Wikipedia. Recent work has accordingly focused on extending DBpedia with additional information from Wikipedia sources like abstracts [3], categories [13], chronology pages [5] and cross-linguistic evidence [8], as well as using statistical inference on the extracted data to derive additional information [10].

In this paper, we build upon this line of work and propose to *extend DBpedia with information mined from Wikipedia's list pages*. Wikipedia, in fact, organizes much of its content within list pages, which are highly structured and can provide strong empirical evidence for building taxonomies, adding information about

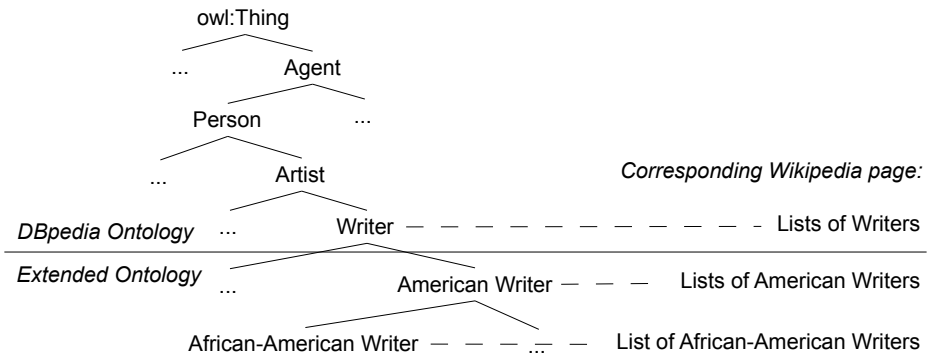


**Fig. 1.** Sample extended ontology built from the Wikipedia list page LISTS OF WRITERS and subordinate list pages

single entities, as well as introducing new entities to DBpedia. As of June 2013, in fact, the English Wikipedia contains 586,346 list pages and 560 lists of lists pages. Apart from being parts of lists of lists, list pages can also be hierarchical, referring to other, finer-grained lists. This creates a certain hierarchy of lists.

List pages usually contain a short introductory text, which describes the topic of the list (e.g., AFRICAN-AMERICAN WRITERS). The main part of the page is a list of links to other Wikipedia pages. This list may be further structured (e.g., alphabetically, chronologically, ...) and contain further information about the entities (for example, birth and death dates of people). In some cases, the list of linked pages is a table with rich information about the entities (cf., for instance, LIST OF SOVEREIGN STATES). Furthermore, links to other pages (other lists or other general pages) may be included in the list page. Accordingly, we show in the following how this kind of pages can turn out to be a goldmine of information to be used to further populate DBpedia.

## 2 Exploiting Wikipedia List Pages for DBpedia

In the following, we describe our methodology to automatically acquire an *extended ontology* for DBpedia from Wikipedia's list pages, and illustrate it through an example focused around the Wikipedia list page LIST OF AFRICAN-AMERICAN WRITERS and its relative list page "generalizations" (e.g., LISTS OF AMERICAN WRITERS, LISTS OF WRITERS, and so on). As shown in Figure 1, our aim is to associate the Wikipedia page LISTS OF WRITERS to the DBpedia class `dbo:Writer`[1], and follow some of its links in order to build a sub-hierarchy of the DBpedia class `dbo:Writer`. That is, in our example, `ext:American_Writer` would be added as a sub-class of the DBpedia ontology class `dbo:Writer` and `ext:African-American_Writer`, in turn, as a subclass of `ext:American_Writer`. Note that, crucially, we are interested in acquiring an extended ontology that contains not only an extended hierarchy, but also defining axioms for the new

[1] We use `dbo` for `http://dbpedia.org/ontology/`, `dbp` for `http://dbpedia.org/resource/`, and `ext` for our own extended ontology.

classes. Thus, for the classes in the above example, we would like to automatically harvest axioms like the following ones:

$$\begin{aligned} ext:American\_Writer &\equiv dbo:Writer \sqcap \\ &\quad \exists\, dbo:nationality.\{dbp:United\_States\} \\ ext:African\_American\_Writer &\equiv ext:American\_Writer \sqcap \\ &\quad \exists\, dbo:ethnicity.\{dbp:African\_American\} \end{aligned}$$

We propose a two-stage process for adding classes to the DBpedia ontology, as well as finding defining axioms. First, frequent patterns are mined from the DBpedia entities corresponding to the list entries, using a framework such as FeGeLOD [11]. These create a list of *candidate classes* and *candidate defining axioms* with *statistical evidence*. Second, *textual evidence* is extracted for the candidates in the list pages' titles and abstracts. Both statistical and textual evidence are finally combined, in order to select suitable (super-)classes in the DBpedia ontology and for selecting the correct defining axioms.

**Adding new classes to the DBpedia ontology.** For this task, we propose to operationalize the two stages as follows:

1. *Find candidate classes using frequent pattern mining.* Since candidate classes in the ontology should be both *frequent* in the set of instances described by the list page, as well as *specific* (in order not to make all extended classes subclasses of `owl:Thing`), measures such as TF-IDF can be used to estimate class relevance. This can be straightforwardly computed from the number of instances in the list page which are typed with a certain DBpedia ontology class (TF), weighted by the inverse frequency of occurrence of that same class as type in the entire DBpedia (IDF).
2. *Weight initial candidate classes using textual evidence.* Type frequency can be further combined with information mined from lexical patterns of the candidate classes in the list pages' titles and abstracts, as well as in their set of linked pages. For instance, we expect most of the DBpedia instances linked from Lists of Writers to be defined as writers in their abstract. Type-denoting patterns could be acquired by exploiting explicit links in the list page, semantifying the abstracts (e.g., using *DBpedia Spotlight* [7]), as well as by identifying generalization-specific verbal patterns (e.g., by means of the *BOA* framework [4]).

**Acquiring new defining axioms for DBpedia ontology classes.** The entities which co-occur in a list typically share certain features. In our example, it is the nationality (American) and the ethnicity (African American). This, in turn, can be used as a source of information to harvest *defining axioms* for the classes derived from the list pages. To this end, we can use the same two-stage approach as for linking the class-denoting list pages to the DBpedia ontology:

1. *Find candidate axioms using frequent pattern mining.* In that case, candidates may not only be simple class expressions, but also property restrictions

such as $\exists\, dbo : ethnicity.\{dbp : African\_American\}$. Again, we argue that TF-IDF – as computed from the frequency of occurrence of properties within the set of instances of a candidate class (e.g., `ext:African-American_Writer`) – can provide useful information in order to select axioms that are both frequent and specific (unlike, e.g., $\exists\, dbo : birthName.\top$, which may be frequent, but rather unspecific).

2. *Weight initial candidate axioms by using textual evidence.* For each candidate axioms, textual evidence is sought in the list page's title and abstract – e.g., African-American writers are mentioned as being African-Americans in the corresponding list page. Similarly to the scenario of class acquisition, we plan to combine statistical and textual evidence, as well as semantify pages' abstracts using tools such as DBpedia Spotlight and BOA.

In the case of `ext:African-American_Writer`, we would like to harvest the following axioms: $dbo : Writer$, $\exists\ dbo : ethnicitiy\{dbp : African\_American\}$, and $\exists\ dbo : nationality\{dbp : United\_States\}$. New defining axioms can then be added as a class definition in the ontology, i.e. $ext : African\_American\_Writer \sqsubseteq ext : American\_Writer \sqcap \exists\ dbo : ethnicitiy\{dbp : African\_American\}$[2].

**Using defining axioms for completing entity data.** For each list page, the identified defining axioms can be added to the respective DBpedia entities for completing the knowledge base, if not yet present. For a list page, all the entities can be augmented with statements corresponding to the defining axioms. In our example, for each entity `x` in the list, the following set of statements would be added, if not yet present:

1. `x dbo:Agent .`
2. `x dbo:Person .`
3. `x dbo:Artist .`
4. `x dbo:Writer .`
5. `x ext:American_Writers .`
6. `x ext:African-American_Writers .`
7. `x dbo:ethnicity dbp:African_American .`
8. `x dbo:nationality dbp:United_States .`

In the end, each entity in the list would carry two more types in the extended ontology, up to four more types in the DBpedia ontology (since DBpedia follows the philosophy of fully materializing super types), and up to two new properties.

Table 1 shows the number of statements that can be created from the Wikipedia list page List of African-American Writers. The newly created axioms include *two entirely new instances* which are not part of the DBpedia knowledge base, since they were derived from red links in the list page. The figures indicate that (i) the number of additional statements that can be created is

[2] Equivalence would be desirable, however, it requires that the list of identified features is complete, and that the defining axioms can be fully expressed in the DBpedia ontology, but both cannot be guaranteed easily. Thus, using subsumption is safer.

**Table 1.** Number of statements that can be created from the Wikipedia page List of African-American Writers.

| Statement | present | missing | TF-IDF |
|---|---|---|---|
| `x a dbpedia-owl:Agent` | 146 | 175 | 0.271 |
| `x a dbpedia-owl:Person` | 150 | 171 | 0.324 |
| `x a dbpedia-owl:Artist` | 94 | 227 | 0.524 |
| `x a dbpedia-owl:Writer` | 80 | 241 | 0.608 |
| *Total type statements* | *470* | *814* | |
| `x dbo:ethnicity dbp:African_American` | 9 | 312 | 0.277 |
| `x dbo:nationality dbp:United_States` | 38 | 283 | 0.127 |
| *Total other axioms* | *47* | *595* | |
| *Total* | *517* | *1409* | |

fairly large – in total, the amount of statements about the instances in the example list page can be enlarged by a factor of almost four, plus the type statements for the extended ontology; (ii) while TF-IDF is a good indicator for identifying classes and defining axioms, additional textual evidence is required. For example, the candidate defining axiom $\exists dbo : deathPlace.\{dbp : New\_York\_City\}$ has a TF-IDF score of 0.157, but it will have a low textual evidence, since New York City is neither mentioned in the list page's title nor its abstract.

## 3 Discussion and Outlook

In this paper, we have introduced an approach for extending DBpedia by exploiting list pages. Our proposed method combines statistical (frequent pattern mining) and textual evidence, obtained with NLP methods. Since there are more than half a million of such list pages in Wikipedia, we can expect them to provide a huge source of valuable information. Accordingly, we have shown that, already in the simple scenario of a single list page, we are able to acquire more than a thousand new statements and add them to DBpedia. As a result, we are able to extend the DBpedia ontology with many detailed classes, as well as axioms for describing them. But while this is indeed good news, we are also aware of the main challenges that lie ahead, in order to make this approach concrete and achieve high precision and wide coverage at the same time:

- First, a labeled *training and evaluation set* of list pages has to be created, which contains both the entities linked from the list page as well as the defining axioms.
- Since list pages are quite diverse (HTML lists with and without sub structures, HTML tables, etc.), *fail-safe methods for extracting the right entities* from the list are essential. This includes the task of distinguishing list items from other hyperlinks found within the list page (e.g., "see also" sections).
- For picking the correct defining axioms, *suitable scoring functions* trading off statistical and textual evidence need to be found. With a training set of labeled list pages, such scoring functions could also be learned with a variety of supervised methods (e.g., regression).

In addition to these challenges, there are also possible extensions of this approach which go beyond the methods described in this paper. Since many list pages use sub-structures which carry some semantics, which could be exploited for adding additional axioms. Likewise, many list pages contain more (structured) information about the linked entities than just links to them (e.g., organize them in tables), which could be extracted as well. Finally, in this paper we focused only on the English version of Wikipedia. Combining evidence from list pages in different language versions would increase both coverage and precision, but comes with challenges of its own.

## References


1. Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia – A crystallization point for the Web of Data. *Web Semantics*, 7(3):154–165, 2009.
2. David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Nico Schlaefer, and Christopher A. Welty. Building Watson: An overview of the DeepQA project. *AI Magazine*, 31(3):59–79, 2010.
3. Aldo Gangemi, Andrea Giovanni Nuzzolese, Valentina Presutti, Francesco Draicchio, Alberto Musetti, and Paolo Ciancarini. Automatic typing of DBpedia entities. In *Proc. of ISWC '12*, pages 65–81, 2012.
4. Daniel Gerber and Axel-Cyrille Ngonga Ngomo. Bootstrapping the linked data web. In *1st ISWC Workshop on Web Scale Knowledge Extraction*, 2011.
5. Daniel Hienert, Daniel Wegener, and Heiko Paulheim. Automatic classification and relationship extraction for multi-lingual and multi-granular events from Wikipedia. In *Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2012)*, volume 902 of *CEUR-WS*, pages 1–10, 2012.
6. Ioana Hulpus, Conor Hayes, Marcel Karnstedt, and Derek Greene. Unsupervised graph-based topic labelling using DBpedia. In *Proc. of WSDM '13*, pages 465–474, 2013.
7. Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*, 2011.
8. Alessio Palmero Aprosio, Claudio Giuliano, and Alberto Lavelli. Automatic expansion of DBpedia exploiting Wikipedia cross-language information. In *10th Extended Semantic Web Conference (ESWC)*, 2013.
9. Heiko Paulheim. Generating possible interpretations for statistics from linked open data. In *9th Extended Semantic Web Conference (ESWC)*, 2012.
10. Heiko Paulheim and Christian Bizer. Type inference on noisy rdf data. In *12th International Semantic Web Conference (ISWC)*, 2013.
11. Heiko Paulheim and Johannes Fürnkranz. Unsupervised Feature Generation from Linked Open Data. In *International Conference on Web Intelligence, Mining, and Semantics (WIMS)*, 2012.
12. Michael Schuhmacher and Simone Paolo Ponzetto. Exploiting DBpedia for web search results clustering. In *Proceedings of the 3rd Workshop on Automatic Knowledge Base Extraction (AKB)*, 2013.
13. Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO: a large ontology from Wikipedia and WordNet. *Journal of Web Semantics*, 6(3):203–217, 2008.