Two Is Bigger (and Better) Than One: the Wikipedia Bitaxonomy Project

Tiziano Flati, Daniele Vannella, Tommaso Pasini and **Roberto Navigli**
Dipartimento di Informatica
Sapienza Università di Roma
{flati,vannella,navigli}@di.uniroma1.it
p.tommaso@gmail.com

Abstract

We present WiBi, an approach to the automatic creation of a bitaxonomy for Wikipedia, that is, an integrated taxonomy of Wikipage pages and categories. We leverage the information available in either one of the taxonomies to reinforce the creation of the other taxonomy. Our experiments show higher quality and coverage than state-of-the-art resources like DBpedia, YAGO, MENTA, WikiNet and WikiTaxonomy. WiBi is available at http://wibitaxonomy.org.

1 Introduction

Knowledge has unquestionably become a key component of current intelligent systems in many fields of Artificial Intelligence. The creation and use of machine-readable knowledge has not only entailed researchers (Mitchell, 2005; Mirkin et al., 2009; Poon et al., 2010) developing huge, broad-coverage knowledge bases (Hovy et al., 2013; Suchanek and Weikum, 2013), but it has also hit big industry players such as Google (Singhal, 2012) and IBM (Ferrucci, 2012), which are moving fast towards large-scale knowledge-oriented systems.

The creation of very large knowledge bases has been made possible by the availability of collaboratively-curated online resources such as Wikipedia and Wiktionary. These resources are increasingly becoming enriched with new content in many languages and, although they are only partially structured, they provide a great deal of valuable knowledge which can be harvested and transformed into structured form (Medelyan et al., 2009; Hovy et al., 2013). Prominent examples include DBpedia (Bizer et al., 2009), BabelNet (Navigli and Ponzetto, 2012), YAGO (Hoffart et al., 2013) and WikiNet (Nastase and Strube, 2013). The types of semantic relation in these resources range from domain-specific, as in Freebase (Bollacker et al., 2008), to unspecified relations, as in BabelNet. However, unlike the case with smaller manually-curated resources such as WordNet (Fellbaum, 1998), in many large automatically-created resources the taxonomical information is either missing, mixed across resources, e.g., linking Wikipedia categories to WordNet synsets as in YAGO, or coarse-grained, as in DBpedia whose hypernyms link to a small upper taxonomy.

Current approaches in the literature have mostly focused on the extraction of taxonomies from the network of Wikipedia categories. WikiTaxonomy (Ponzetto and Strube, 2007), the first approach of this kind, is based on the use of heuristics to determine whether is-a relations hold between a category and its subcategories. Subsequent approaches have also exploited heuristics, but have extended them to any kind of semantic relation expressed in the category names (Nastase and Strube, 2013). But while the aforementioned attempts provide structure for categories that supply meta-information for Wikipedia pages, surprisingly little attention has been paid to the acquisition of a full-fledged taxonomy for Wikipedia pages themselves. For instance, Ruiz-Casado et al. (2005) provide a general vector-based method which, however, is incapable of linking pages which do not have a WordNet counterpart. Higher coverage is provided by de Melo and Weikum (2010) thanks to the use of a set of effective heuristics, however, the approach also draws on WordNet and sense frequency information.

In this paper we address the task of taxonomizing Wikipedia in a way that is fully independent of other existing resources such as WordNet. We present WiBi, a novel approach to the creation of a Wikipedia bitaxonomy, that is, a taxonomy of Wikipedia pages aligned to a taxonomy of categories. At the core of our approach lies the idea that the information at the page and category

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 945–955,
Baltimore, Maryland, USA, June 23-25 2014.

level are mutually beneficial for inducing a wide-coverage and fine-grained integrated taxonomy.

2 WiBi: A Wikipedia Bitaxonomy

We induce a Wikipedia bitaxonomy, i.e., a taxonomy of pages and categories, in 3 phases:

1. **Creation of the initial page taxonomy:** we first create a taxonomy for the Wikipedia pages by parsing textual definitions, extracting the hypernym(s) and disambiguating them according to the page inventory.
2. **Creation of the bitaxonomy:** we leverage the hypernyms in the page taxonomy, together with their links to the corresponding categories, so as to induce a taxonomy over Wikipedia categories in an iterative way. At each iteration, the links in the page taxonomy are used to identify category hypernyms and, conversely, the new category hypernyms are used to identify more page hypernyms.
3. **Refinement of the category taxonomy:** finally we employ structural heuristics to overcome inherent problems affecting categories.

The output of our three-phase approach is a bitaxonomy of millions of pages and hundreds of thousands of categories for the English Wikipedia.

3 Phase 1: Inducing the Page Taxonomy

The goal of the first phase is to induce a taxonomy of Wikipedia pages. Let P be the set of all the pages and let $T_P = (P, E)$ be the page taxonomy directed graph whose nodes are pages and whose edge set E is initially empty ($E := \emptyset$). For each $p \in P$ our aim is to identify the most suitable generalization $p_h \in P$ so that we can create the edge (p, p_h) and add it to E. For instance, given the page APPLE, which represents the fruit meaning of *apple*, we want to determine that its hypernym is FRUIT and add the hypernym edge connecting the two pages (i.e., $E := E \cup \{(\text{APPLE}, \text{FRUIT})\}$). To do this, we perform a syntactic step, in which the hypernyms are extracted from the page's textual definition, and a semantic step, in which the extracted hypernyms are disambiguated according to the Wikipedia inventory.

3.1 Syntactic step: hypernym extraction

In the syntactic step, for each page $p \in P$, we extract zero, one or more hypernym lemmas, that is, we output potentially ambiguous hypernyms for the page. The first assumption, which follows

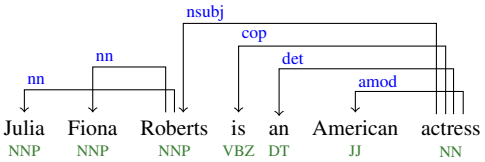


Figure 1: A dependency tree example with copula.

the Wikipedia guidelines and is validated in the literature (Navigli and Velardi, 2010; Navigli and Ponzetto, 2012), is that the first sentence of each Wikipedia page p provides a textual definition for the concept represented by p. The second assumption we build upon is the idea that a lexical taxonomy can be obtained by extracting hypernyms from textual definitions. This idea dates back to the early 1970s (Calzolari et al., 1973), with later developments in the 1980s (Amsler, 1981; Calzolari, 1982) and the 1990s (Ide and Véronis, 1993).

To extract hypernym lemmas, we draw on the notion of copula, that is, the relation between the complement of a copular verb and the copular verb itself. Therefore, we apply the Stanford parser (Klein and Manning, 2003) to the definition of a page in order to extract all the dependency relations of the sentence. For example, given the definition of the page JULIA ROBERTS, i.e., "Julia Fiona Roberts is an American actress.", the Stanford parser outputs the set of dependencies shown in Figure 1. The noun involved in the copula relation is *actress* and thus it is taken as the page's hypernym lemma. However, the extracted hypernym is sometimes overgeneral (*one*, *kind*, *type*, etc.). For instance, given the definition of the page APOLLO, "Apollo is one of the most important and complex of the Olympian deities in ancient Greek and Roman religion [...].", the only copula relation extracted is between *is* and *one*. To cope with this problem we use a list of stopwords.[1]When such a term is extracted as hypernym, we replace it with the rightmost noun of the first following noun sequence (e.g., *deity* in the above example). If the resulting lemma is again a stopword we repeat the procedure, until a valid hypernym or no appropriate hypernym can be found. Finally, to capture multiple hypernyms, we iteratively follow the *conj_and* and *conj_or* relations starting from the initially extracted hypernym. For example, consider the definition of ARISTOTLE: "Aristotle was a Greek philosopher and polymath, a student of Plato and teacher of Alexander the Great." Initially, the *philosopher* hypernym is selected thanks to the copula relation, then, fol-

[1]E.g., *species*, *genus*, *one*, etc. Full list available online.

lowing the conjunction relations, also *polymath*, *student* and *teacher* are extracted as hypernyms. While more sophisticated approaches like Word-Class Lattices could be applied (Navigli and Velardi, 2010), we found that, in practice, our hypernym extraction approach provides higher coverage, which is critical in our case.

3.2 Semantic step: hypernym disambiguation

Since our aim is to connect pairs of pages via hypernym relations, our second step consists of disambiguating the obtained hypernym lemmas of page p by associating the most suitable page with each hypernym. Following previous work (Ruiz-Casado et al., 2005; Navigli and Ponzetto, 2012), as the inventory for a given lemma we consider the set of pages whose main title is the lemma itself, except for the sense specification in parenthesis. For instance, given *fruit* as the hypernym for APPLE we would like to link APPLE to FRUIT as opposed to, e.g., FRUIT (BAND) or FRUIT (ALBUM).

3.2.1 Hypernym linkers

To disambiguate hypernym lemmas, we exploit the structural features of Wikipedia through a pipeline of hypernym linkers $\mathcal{L} = \{L_i\}$, applied in cascade order (cf. Section 3.3.1). We start with the set of page-hypernym pairs $H = \{(p, h)\}$ as obtained from the syntactic step. The successful application of a linker to a pair $(p, h) \in H$ yields a page p_h as the most suitable sense of h, resulting in setting $isa(p, h) = p_h$. At step i, the i-th linker $L_i \in \mathcal{L}$ is applied to H and all the hypernyms which the linker could disambiguate are removed from H. This prevents lower-precision linkers from overriding decisions taken by more accurate ones.

We now describe the hypernym linkers. In what follows we denote with $p \xrightarrow{h} p_h$ the fact that the definition of a Wikipedia page p contains an occurrence of h linked to page p_h. Note that p_h is not necessarily a sense of h.

Crowdsourced linker If $p \xrightarrow{h} p_h$, i.e., the hypernym h is found to have been manually linked to p_h in p by Wikipedians, we assign $isa(p, h) = p_h$. For example, because *capital* was linked in the BRUSSELS page definition to CAPITAL CITY, we set isa(BRUSSELS, *capital*) = CAPITAL CITY.

Category linker Given the set $W \subset P$ of Wikipedia pages which have at least one category in common with p, we select the majority sense of h, if there is one, as hyperlinked across all the definitions of pages in W:

$$isa(p, h) = \arg\max_{p_h} \sum_{p' \in W} 1(p' \xrightarrow{h} p_h)$$

where $1(p' \xrightarrow{h} p_h)$ is the characteristic function which equals 1 if h is linked to p_h in page p', 0 otherwise. For example, the linker sets isa(EGGPLANT, *plant*) = PLANT because most of the pages associated with TROPICAL FRUIT, a category of EGGPLANT, contain in their definitions the term *plant* linked to the PLANT page.

Multiword linker If $p \xrightarrow{m} p_h$ and m is a multiword expression containing the lemma h as one of its words, set $isa(p, h) = p_h$. For example, we set isa(PROTEIN, *compound*) = CHEMICAL COMPOUND, as *chemical compound* is linked to CHEMICAL COMPOUND in the definition of PROTEIN.

Monosemous linker If h is monosemous in Wikipedia (i.e., there is only a single page p_h for that lemma), link it to its only sense by setting $isa(p, h) = p_h$. For example, we extract the hypernym *businessperson* from the definition of MERCHANT and, as it is unambiguous, we link it to BUSINESSPERSON.

Distributional linker Finally, we provide a distributional approach to hypernym disambiguation. We represent the textual definition of page p as a distributional vector $\vec{v}_p$ whose components are all the English lemmas in Wikipedia. The value of each component is the occurrence count of the corresponding content word in the definition of p.

The goal of this approach is to find the best link for hypernym h of p among the pages h is linked to, across the whole set of definitions in Wikipedia. Formally, for each p_h such that h is linked to p_h in some definition, we define the set of pages $P(p_h)$ whose definitions contain a link to p_h, i.e., $P(p_h) = \{p' \in P | p' \xrightarrow{h} p_h\}$. We then build a distributional vector $\vec{v}_{p'}$ for each $p' \in P(p_h)$ as explained above and create an aggregate vector $\vec{v}_{p_h} = \sum_{p'} \vec{v}_{p'}$. Finally, we determine the similarity of p to each p_h by calculating the dot product between the two vectors $sim(p, p_h) = \vec{v}_p \cdot \vec{v}_{p_h}$. If $sim(p, p_h) > 0$ for any p_h we perform the following association:

$$isa(p, h) = \arg\max_{p_h} sim(p, p_h)$$

For example, thanks to this linker we set isa(VACUUM CLEANER, *device*) = MACHINE.

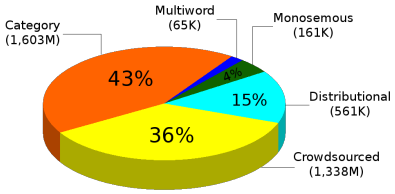


Figure 2: Distribution of linked hypernyms.

3.3 Page Taxonomy Evaluation

Statistics We applied the above linkers to the October 2012 English Wikipedia dump. Out of the 3,829,058 total pages, 4,270,232 hypernym lemmas were extracted in the syntactic step for 3,697,113 pages (covering more than 96% of the total). Due to illformed definitions, though, it was not always possible to extract the hypernym lemma: for example, 6 APRIL 2010 BAGHDAD BOMBINGS is defined as "A series of bomb explosions destroyed several buildings in Baghdad", which only implicitly provides the hypernym.

The semantic step disambiguated 3,718,612 hypernyms for 3,294,562 Wikipedia pages, i.e., covering more than 86% of the English pages with at least one disambiguated hypernym. Figure 2 plots the number and distribution of hypernyms disambiguated by our hypernym linkers.

Taxonomy quality To evaluate the quality of our page taxonomy we randomly sampled 1,000 Wikipedia pages. For each page we provided: i) a list of suitable hypernym lemmas for the page, mainly selected from its definition; ii) for each lemma the correct hypernym page(s). We calculated precision as the average ratio of correct hypernym lemmas (senses) to the total number of lemmas (senses) returned for all the pages in the dataset, recall as the number of correct lemmas (senses) over the total number of lemmas (senses) in the dataset, and coverage as the fraction of pages for which at least one lemma (sense) was returned, independently of its correctness. Results, both at lemma- and sense-level, are reported in Table 1. Not only does our taxonomy show high precision and recall in extracting ambiguous hypernyms, it also disambiguates more than 3/4 of the hypernyms with high precision.

3.3.1 Hypernym linker order

The optimal order of application of the above linkers is the same as that presented in Section 3.2.1. It was established by selecting the combination, among all possible permutations, which maximized precision on a tuning set of 100 randomly sampled pages, disjoint from our page dataset.

	Prec.	Rec.	Cov.
Lemma	94.83	90.20	98.50
Sense	82.77	75.10	89.20

Table 1: Page taxonomy performance.

4 Phase 2: Inducing the Bitaxonomy

The page taxonomy built in Section 3 will serve as a stable, pivotal input to the second phase, the aim of which is to build our bitaxonomy, that is, a taxonomy of pages and categories. Our key idea is that the generalization-specialization information available in each of the two taxonomies is mutually beneficial. We implement this idea by exploiting one taxonomy to update the other, and vice versa, in an iterative way, until a fixed point is reached. The final output of this phase is, on the one hand, a page taxonomy augmented with additional hypernymy relations and, on the other hand, a category taxonomy which is built from scratch.

4.1 Initialization

Our bitaxonomy $B = \{T_P, T_C\}$ is a pair consisting of the page taxonomy $T_P = (P, E)$, as obtained in Section 3, and the category taxonomy $T_C = (C, \emptyset)$, which initially contains all the categories as nodes but does not include any hypernym edge between category nodes. In the following we describe the core algorithm of our approach, which iteratively and mutually populates and refines the edge sets $E(T_P)$ and $E(T_C)$.

4.2 The Bitaxonomy Algorithm

Preliminaries Before proceeding, we define some basic concepts that will turn out to be useful for presenting our algorithm. We denote by $super_T(t)$ the set of all ancestors of a node t in the taxonomy T (be it T_P or T_C). We further define a verification function $t \rightsquigarrow_T t'$ which, in the case of T_C, returns true if there is a path in the Wikipedia category network between t and t', false otherwise, and, in the case of T_P, returns true if t' is a sense, i.e., a page, of a hypernym h of t (that is, $(t, h) \in H$, cf. Section 3.2.1). For instance, SPORTSMEN $\rightsquigarrow_{T_C}$ MEN BY OCCUPATION holds for categories because the former is a sub-category of the latter in Wikipedia, and RADIOHEAD $\rightsquigarrow_{T_P}$ BAND (MUSIC) for pages, because *band* is a hypernym extracted from the textual definition of RADIOHEAD and BAND (MUSIC) is a sense of *band* in Wikipedia. Note that, while the *super* function returns information that we have already learned, i.e., it is in T_P and T_C, the $\rightsquigarrow$ operator

holds just for candidate is-a relations, as it uses knowledge from Wikipedia itself which is potentially incorrect. For instance, SPORTSMEN $\rightsquigarrow_{T_C}$ MEN'S SPORTS in the Wikipedia category network, and RADIOHEAD $\rightsquigarrow_{T_P}$ BAND (RADIO) between the two Wikipedia pages, both hold according to our definition of $\rightsquigarrow$, while connecting the wrong hypernym candidates. At the core of our algorithm, explained below, is the mutual leveraging of the *super* function from one of the two taxonomies (pages or categories) to decide about which candidates (for which a $\rightsquigarrow$ relation holds) in the other taxonomy are real hypernyms.

Finally, we define the projection operator π, such that $\pi(c)$, $c \in C$, is the set of pages categorized with c, and $\pi(p)$, $p \in P$, is the set of categories associated with page p in Wikipedia. For instance, the pages which belong to the category OLYMPIC SPORTS are given by π(OLYMPIC SPORTS) = {BASEBALL, BOXING, ..., TRIATHLON}. Vice versa, π(TRIATHLON) = {MULTISPORTS, OLYMPIC SPORTS, ..., OPEN WATER SWIMMING}. The projection operator π enables us to jump from one taxonomy to the other and expresses the mutual membership relation between pages and categories.

Algorithm We now describe in detail the bitaxonomy algorithm, whose pseudocode is given in Algorithm 1. The algorithm takes as input the two taxonomies, initialized as described in Section 4.1. Starting from the category taxonomy (line 1), the algorithm updates the two taxonomies in turn, until convergence is reached, i.e., no more edges can be added to any side of the bitaxonomy. Let T be the current taxonomy considered at a given moment and T' its dual taxonomy. The algorithm proceeds by selecting a node $t \in V(T)$ for which no hypernym edge (t, t_h) could be found up until that moment (line 3), and then tries to infer such a relation by drawing on the dual taxonomy T' (lines 5-12). This is the core of the bitaxonomy algorithm, in which hypernymy knowledge is transferred from one taxonomy to the other. By applying the projection operator π to t, the algorithm considers those nodes t' aligned to t in the dual taxonomy (line 5) and obtains their hypernyms t'_h using the $super_{T'}$ function (line 6). The nodes reached in T' act as a clue for discovering the suitable hypernyms for the starting node $t \in V(T)$. To perform the discovery, the algorithm projects each such hypernym node $t'_h \in S$ and increments the count of each projection $t_h \in \pi(t'_h)$ (line 8). Finally, the node $\hat{t}_h \in V(T)$ with maximum count, and such that $t \rightsquigarrow_T \hat{t}_h$ holds, if one exists, is promoted as hypernym of t and a new hypernym edge $(t, \hat{t}_h)$ is added to $E(T)$ (line 12). Finally, the role of the two taxonomies is swapped and the process is repeated until no more change is possible.

Algorithm 1 The Bitaxonomy Algorithm

Input: T_P, T_C
1: $T := T_C, T' := T_P$
2: **repeat**
3: **for all** $t \in V(T)$ s.t. $\nexists(t, t_h) \in E(T)$ **do**
4: **reset** *count*
5: **for all** $t' \in \pi(t)$ **do**
6: $S := super_{T'}(t')$
7: **for all** $t'_h \in S$ **do**
8: **for all** $t_h \in \pi(t'_h)$ **do** $count(t_h)$++ **end for**
9: **end for**
10: **end for**
11: $\hat{t}_h := \arg\max_{t_h:\, t \rightsquigarrow_T t_h} count(t_h)$
12: **if** $count(\hat{t}_h) > 0$ **then** $E(T) := E(T) \cup \{(t, \hat{t}_h)\}$
13: **end for**
14: **swap** T and T'
15: **until** convergence
16: **return** $\{T, T'\}$

Example Let us illustrate the algorithm by way of an example. Assume we are in the first iteration ($T = T_C$) and consider the Wikipedia category t = OLYMPICS (line 3) and its super-categories {MULTI-SPORT EVENTS, SPORT AND POLITICS, INTERNATIONAL SPORTS COMPETITIONS}. This category has 27 pages associated with it (line 5), 23 of which provide a hypernym page in T_P (line 6): e.g., PARALYMPIC GAMES, associated with the category OLYMPICS, is a MULTI-SPORT EVENT and is therefore contained in S. By considering and counting the categories of each page in S (lines 7-8), we end up counting the category MULTI-SPORT EVENTS four times and other categories, such as AWARDS and SWIMSUITS, once. As MULTI-SPORT EVENTS has the highest count and is connected to OLYMPICS by a path in the Wikipedia category network (line 11), the hypernym edge (OLYMPICS, MULTI-SPORT EVENTS) is added to T_C (line 12).

5 Phase 3: Category taxonomy refinement

As the final phase, we refine and enrich the category taxonomy. The goal of this phase is to provide broader coverage to the category taxonomy T_C created as explained in Section 4. We apply three enrichment heuristics which add hypernyms to those categories c for which no hypernym could be found in phase 2, i.e., $\nexists c'$ s.t. $(c, c') \in E(T_C)$.

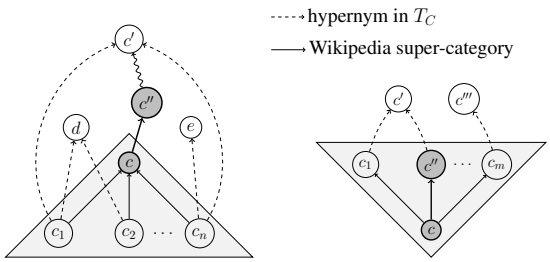
Single super-category As a first structural refinement, we automatically link an uncovered category c to c' if c' is the only direct super-category of c in Wikipedia.

Sub-categories We increase the hypernym coverage by exploiting the sub-categories of each uncovered category c (see Figure 3a). In detail, for each uncovered category c we consider the set $sub(c)$ of all the Wikipedia subcategories of c (nodes $c_1, c_2, \ldots, c_n$ in Figure 3a) and then let each category vote, according to its direct hypernym categories in T_C (the vote is as in Algorithm 1). Then we proceed in decreasing order of vote and select the highest-ranking category c' which is connected to c via a path in T_C, i.e., $c \rightsquigarrow_{T_C} c'$. We then pick up the direct ancestor c'' of c which lies in the path from c to c' and add the hypernym edge (c, c'') to $E(T_C)$. For example, consider the category FRENCH TELEVISION PEOPLE; since this category has no associated pages, in phase 2 no hypernym could be found. However, by applying the sub-categories heuristic, we discover that TELEVISION PEOPLE BY COUNTRY is the hypernym most voted by our target category's descendants, such as FRENCH TELEVISION ACTORS and FRENCH TELEVISION DIRECTORS. Since TELEVISION PEOPLE BY COUNTRY is at distance 1 in the Wikipedia category network from FRENCH TELEVISION PEOPLE, we add (FRENCH TELEVISION PEOPLE, TELEVISION PEOPLE BY COUNTRY) to $E(T_C)$.

Super-categories We then apply a similar heuristic involving super-categories (see Figure 3b). Given an uncovered category c, we consider its direct Wikipedia super-categories and let them vote, according to their hypernym categories in T_C. Then we proceed in decreasing order of vote and select the highest-ranking category c' which is connected to c in T_C, i.e., $c \rightsquigarrow_{T_C} c'$. We then pick up the direct ancestor c'' of c which lies in the path from c to c' and add the edge (c, c'') to $E(T_C)$.

5.1 Bitaxonomy Evaluation

Category taxonomy statistics We applied phases 2 and 3 to the output of phase 1, which was evaluated in Section 3.3. In Figure 4a we show the increase in category coverage at each iteration throughout the execution of the two phases (1SUP, SUB and SUPER correspond to the three above heuristics of phase 3). The final outcome is a category taxonomy which includes 594,917 hypernymy links between categories,



(a) Sub categ. heuristic. (b) Super categ. heuristic.

Figure 3: Heuristic patterns for the coverage refinement of the category taxonomy.

covering more than 96% of the 618,641 categories in the October 2012 English Wikipedia dump. The graph shows the steepest slope in the first iterations of phase 2, which converges around 400k categories at iteration 30, and a significant boost due to phase 3 producing another 175k hypernymy edges, with the super-category heuristic contributing most. 78.90% of the nodes in T_C belong to the same connected component. The average height of the biggest component of T_C is 23.26 edges and the maximum height is 49. We note that the average height of T_C is much greater than that of T_P, which reflects the category taxonomy distinguishing between very subtle classes, such as ALBUMS BY ARTISTS, ALBUMS BY RECORDING LOCATION, etc.

Category taxonomy quality To estimate the quality of the category taxonomy, we randomly sampled 1,000 categories and, for each of them, we manually associated the super-categories which were deemed to be appropriate hypernyms. Figure 4b shows the performance trend as the algorithm iteratively covers more and more categories. Phase 2 is particularly robust across iterations, as it leads to increased recall while retaining very high precision. As regards phase 3, the super-categories heuristic leads to only a slight precision decrease, while improving recall considerably. Overall, the final taxonomy T_C achieves 85.80% precision, 83.40% recall and 97.20% coverage on our dataset.

Page taxonomy improvement As a result of phase 2, 141,105 additional hypernymy links were also added to the page taxonomy, resulting in an overall 82.99% precision, 77.90% recall and 92.10% coverage, with a non-negligible 3% boost from phase 1 to phase 2 in terms of recall and coverage on our Wikipedia page dataset.

We also calculated some statistics for the resulting taxonomy obtained by aggregating the 3.8M

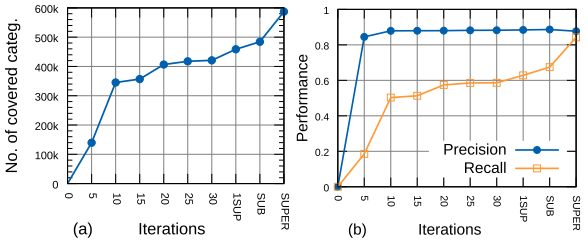


Figure 4: Category taxonomy evaluation.

hypernym links in a single directed graph. Overall, 99% of nodes belong to the same connected component, with a maximum height of 29 and an average height on the biggest component of 6.98.

6 Related Work

Although the extraction of taxonomies from machine-readable dictionaries was already being studied in the early 1970s (Calzolari et al., 1973), pioneering work on large amounts of data only appeared in the 1990s (Hearst, 1992; Ide and Véronis, 1993). Approaches based on handcrafted patterns and pattern matching techniques have been developed to provide a supertype for the extracted terms (Etzioni et al., 2004; Blohm, 2007; Kozareva and Hovy, 2010; Navigli and Velardi, 2010; Velardi et al., 2013, inter alia). However, these methods do not link terms to existing knowledge resources such as WordNet, whereas those that explicitly link do so by adding new leaves to the existing taxonomy instead of acquiring wide-coverage taxonomies from scratch (Pantel and Ravichandran, 2004; Snow et al., 2006).

The recent upsurge of interest in collaborative knowledge curation has enabled several approaches to large-scale taxonomy acquisition (Hovy et al., 2013). Most approaches initially focused on the Wikipedia category network, an entangled set of generalization-containment relations between Wikipedia categories, to extract the hypernymy taxonomy as a subset of the network. The first approach of this kind was WikiTaxonomy (Ponzetto and Strube, 2007; Ponzetto and Strube, 2011), based on simple, yet effective lightweight heuristics, totaling more than 100k is-a relations. Other approaches, such as YAGO (Suchanek et al., 2008; Hoffart et al., 2013), yield a taxonomical backbone by linking Wikipedia categories to WordNet. However, the categories are linked to the first, i.e., most frequent, sense of the category head in WordNet, involving only leaf categories in the linking.

Interest in taxonomizing Wikipedia pages, instead, developed with DBpedia (Auer et al., 2007), which pioneered the current stream of work aimed at extracting semi-structured information from Wikipedia templates and infoboxes. In DBpedia, entities are mapped to a coarse-grained ontology which is collaboratively maintained and contains only about 270 classes corresponding to popular named entity types, in contrast to our goal of structuring the full set of Wikipedia articles in a larger and finer-grained taxonomy.

A few notable efforts to reconcile the two sides of Wikipedia, i.e., pages and categories, have been put forward very recently: WikiNet (Nastase et al., 2010; Nastase and Strube, 2013) is a project which heuristically exploits different aspects of Wikipedia to obtain a multilingual concept network by deriving not only is-a relations, but also other types of relations. A second project, MENTA (de Melo and Weikum, 2010), creates one of the largest multilingual lexical knowledge bases by interconnecting more than 13M articles in 271 languages. In contrast to our work, hypernym extraction is supervised in that decisions are made on the basis of labelled training examples and requires a reconciliation step owing to the heterogeneous nature of the hypernyms, something that we only do for categories, due to their noisy network. While WikiNet and MENTA bring together the knowledge available both at the page and category level, like we do, they either achieve low precision and coverage of the taxonomical structure or exhibit overly general hypernyms, as we show in our experiments in the next section.

Our work differs from the others in at least three respects: first, in marked contrast to most other resources, but similarly to WikiNet and WikiTaxonomy, our resource is self-contained and does not depend on other resources such as WordNet; second, we address the taxonomization task on both sides, i.e., pages and categories, by providing an algorithm which mutually and iteratively transfers knowledge from one side of the bitaxonomy to the other; third, we provide a wide coverage bitaxonomy closer in structure and granularity to a manual WordNet-like taxonomy, in contrast, for example, to DBpedia's flat entity-focused hierarchy.[2]

[2]Note that all the competitors on categories have average height between 1 and 3.69 on their biggest component, while we have 23.26, while on pages their height is between 1.9 and 4.22, while ours is 6.98. Since WordNet's average height is 8.07 we deem WiBi to be the resource structurally closest to WordNet.

Dataset	System	Prec.	Rec.	Cov.
Pages	WiBi	84.11	79.40	92.57
	WikiNet	57.29††	71.45††	82.01
	DBpedia	87.06	51.50††	55.93
	MENTA	81.52	72.49†	88.92
Categories	WiBi	85.18	82.88	97.31
	WikiTax	88.50	54.83††	59.43
	YAGO	94.13	53.41††	56.74
	MENTA	87.11	84.63	97.15
	MENTA$^{-\text{ENT}}$	85.18	71.95††	84.47

Table 2: Page and category taxonomy evaluation. † (††) denotes statistically significant difference, using χ^2 test, $p < 0.02$ ($p < 0.01$) between WiBi and the daggered resource.

7 Comparative Evaluation

7.1 Experimental Setup

We compared our resource (WiBi) against the Wikipedia taxonomies of the major knowledge resources in the literature providing hypernym links, namely DBpedia, WikiNet, MENTA, WikiTaxonomy and YAGO (see Section 6). As datasets, we used our gold standards of 1,000 randomly-sampled pages (see Section 3.3) and categories (see Section 5.1). In order to ensure a level playing field, we detected those pages (categories) which do not exist in any of the above resources and removed them to ensure full coverage of the dataset across all resources. For each resource we calculated precision, by manually marking each hypernym returned for each page (category) as correct or not. As regards recall, we note that in two cases (i.e., DBpedia returning page supertypes from its upper taxonomy, YAGO linking categories to WordNet synsets) the generalizations are neither pages nor categories and that MENTA returns heterogeneous hypernyms as mixed sets of WordNet synsets, Wikipedia pages and categories. Given this heterogeneity, standard recall across resources could not be calculated. For this reason we calculated recall as described in Section 3.3.

7.2 Results

Wikipedia pages We first report the results of the knowledge resources which provide page hypernyms, i.e., we compare against WikiNet, DBpedia and MENTA. We use the original outputs from the three resources: the first two are based on dumps which are from the same year as the one used in WiBi (cf. Section 3.3), while MENTA is based on a dump dating back to 2010 (consisting of 3.25M pages and 565k categories). We decided to include the latter for comparison purposes, as it uses knowledge from 271 Wikipedias to build the final taxonomy. However, we recognize its performance might be relatively higher on a 2012 dump.

We show the results on our page hypernym dataset in Table 2 (top). As can be seen, WikiNet obtains the lowest precision, due to the high number of hypernyms provided, many of which are incorrect, with a recall between that of DBpedia and MENTA. WiBi outperforms all other resources with 84.11% precision, 79.40% recall and 92.57% coverage. MENTA seems to be the closest resource to ours, however, we remark that the hypernyms output by MENTA are very heterogeneous: 48% of answers are represented by a WordNet synset, 37% by Wikipedia categories and 15% are Wikipedia pages. In contrast to all other resources, WiBi outputs page hypernyms only.

Wikipedia categories We then compared all the knowledge resources which deal with categories, i.e., WikiTaxonomy, YAGO and MENTA. For the latter two, the above considerations about the 2012 dump hold, whereas we reimplemented WikiTaxonomy, which was based on a 2009 dump, to run it on the same dump as WiBi. We excluded WikiNet from our comparison because it turned out to have low coverage of categories (i.e., less than 1%).

We show the results on our category dataset in Table 2 (bottom). Despite other systems exhibiting higher precision, WiBi generally achieves higher recall, thanks also to its higher category coverage. YAGO obtains the lowest recall and coverage, because only leaf categories are considered. MENTA is the closest system to ours, obtaining slightly higher precision and recall. Notably, however, MENTA outputs the first WordNet sense of *entity* for 13.17% of all the given answers, which, despite being correct and accounted in precision and recall, is uninformative. Since a system which always outputs *entity* would maximise all the three measures, we also calculated the performance for MENTA when discarding *entity* as an answer; as Table 2 shows (bottom, MENTA$^{-\text{ENT}}$), recall drops to 71.95%. Further analysis, presented below, shows that the specificity of its hypernyms is considerably lower than that of WiBi.

7.3 Analysis of the results

To get further insight into our results we performed two additional analyses of the data. First, we estimated the level of specialization of the hypernyms in the different resources on our two datasets. The idea is that a hypernym should be

Dataset	System (X)	WiBi=X	WiBi>X	WiBi<X
Pages	WikiNet	33.38	34.94	31.68
	DBpedia	31.68	56.71	11.60
	MENTA	19.04	50.85	30.12
Categories	WikiTax	43.11	38.51	18.38
	YAGO	12.36	81.14	6.50
	MENTA	12.36	73.69	13.95

Table 3: Specificity comparison.

valid while at the same time being as specific as possible (e.g., SINGER should be preferred over PERSON). We therefore calculated a measure, which we called specificity, that computes the percentage of times a system outputs a more specific answer than another system. To do this, we annotated each hypernym returned by a system as follows: -1 if the answer was wrong, 0 if missing, $>$ 0 if correct; more specific answers were assigned higher scores. When comparing two systems, we select the respective most specific answers a_1, a_2 and say the first system is more specific than the latter whenever $score(a_1) > score(a_2)$. Table 3 shows the results for all the resources and for both the page and category taxonomies: WiBi consistently provides considerably more specific hypernyms than any other resource (middle column).

A second important aspect that we analyzed was the granularity of each taxonomy, determined by drawing each resource on a bidimensional plane with the number of distinct hypernyms on the x axis and the total number of hypernyms (i.e., edges) in the taxonomy on the y axis. Figures 5a and 5b show the position of each resource for the page and the category taxonomies, respectively. As can be seen, WiBi, as well as the page taxonomy of MENTA, is the resource with the best granularity, as not only does it attain high coverage, but it also provides a larger variety of classes as generalizations of pages and categories. Specifically, WiBi provides over 3M hypernym pages chosen from a range of 94k distinct hypernyms, while others exhibit a considerably smaller range of distinct hypernyms (e.g., DBpedia by design, but also WikiNet, with around 11k distinct page hypernyms). The large variety of classes provided by MENTA, however, is due to including more than 100k Wikipedia categories (among which, categories about *deaths* and *births* alone represent about 2% of the distinct hypernyms). As regards categories, while the number of distinct hypernyms of WiBi and WikiTaxonomy is approximately the same (around 130k), the total number of hypernyms (around 580k for both taxonomies) is distributed over half of the categories in WikiTaxonomy compared to WiBi, resulting in a double number of hypernyms per category, but lower coverage (cf. Table 2).

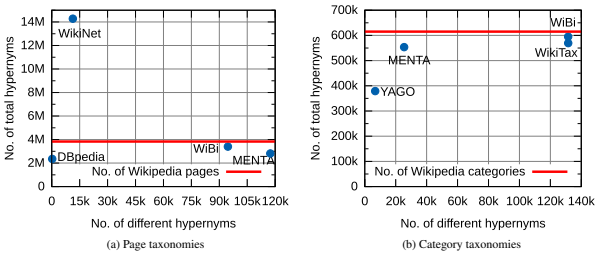


Figure 5: Hypernym granularity for the resources.

8 Conclusions

In this paper we have presented WiBi, an automatic 3-phase approach to the construction of a bitaxonomy for the English Wikipedia, i.e., a full-fledged, integrated page and category taxonomy: first, using a set of high-precision linkers, the page taxonomy is populated; next, a fixed point algorithm populates the category taxonomy while enriching the page taxonomy iteratively; finally, the category taxonomy undergoes structural refinements. Coverage, quality and granularity of the bitaxonomy are considerably higher than the taxonomy of state-of-the-art resources like DBpedia, YAGO, MENTA, WikiNet and WikiTaxonomy.

Our contributions are three-fold: i) we propose a unified, effective approach to the construction of a Wikipedia bitaxonomy, a richer structure than those produced in the literature; ii) our method for building the bitaxonomy is self-contained, thanks to its independence from external resources (like WordNet) and the virtual absence of supervision, making WiBi replicable on any new version of Wikipedia; iii) the taxonomy provides nearly full coverage of pages and categories, encompassing the entire encyclopedic knowledge in Wikipedia.

We will apply our video games with a purpose (Vannella et al., 2014) to validate WiBi. We also plan to integrate WiBi into BabelNet (Navigli and Ponzetto, 2012), so as to fully taxonomize it, and exploit its high quality for improving semantic predicates (Flati and Navigli, 2013).

Acknowledgments

The authors gratefully acknowledge the support of the ERC Starting Grant MultiJEDI No. 259234.
We thank Luca Telesca for his implementation of WikiTaxonomy and Jim McManus for his comments on the manuscript.

References

Robert A. Amsler. 1981. A Taxonomy for English Nouns and Verbs. In *Proceedings of Association for Computational Linguistics (ACL '81)*, pages 133–138, Stanford, California, USA.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ive. 2007. DBpedia: A nucleus for a web of open data. In *Proceedings of 6th International Semantic Web Conference joint with 2nd Asian Semantic Web Conference (ISWC+ASWC 2007)*, pages 722–735, Busan, Korea.

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. DBpedia - a crystallization point for the Web of Data. *Web Semantics*, 7(3):154–165.

Sebastian Blohm. 2007. Using the web to reduce data sparseness in pattern-based information extraction. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 18–29, Warsaw, Poland. Springer.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the International Conference on Management of Data (SIGMOD '08)*, SIGMOD '08, pages 1247–1250, New York, NY, USA.

Nicoletta Calzolari, Laura Pecchia, and Antonio Zampolli. 1973. Working on the Italian Machine Dictionary: a Semantic Approach. In *Proceedings of the 5th Conference on Computational Linguistics (COLING '73)*, pages 49–70, Pisa, Italy.

Nicoletta Calzolari. 1982. Towards the organization of lexical definitions on a database structure. In *Proc. of the 9th Conference on Computational Linguistics (COLING '82)*, pages 61–64, Prague, Czechoslovakia.

Gerard de Melo and Gerhard Weikum. 2010. MENTA: Inducing Multilingual Taxonomies from Wikipedia. In *Proceedings of Conference on Information and Knowledge Management (CIKM '10)*, pages 1099–1108, New York, NY, USA.

Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2004. Web-scale information extraction in knowItAll: (preliminary results). In *Proceedings of the 13th International Conference on World Wide Web (WWW '04)*, pages 100–110, New York, NY, USA. ACM.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.

David A. Ferrucci. 2012. Introduction to "This is Watson". *IBM Journal of Research and Development*, 56(3):1.

Tiziano Flati and Roberto Navigli. 2013. SPred: Large-scale Harvesting of Semantic Predicates. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1222–1232, Sofia, Bulgaria.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the International Conference on Computational Linguistics (COLING '92)*, pages 539–545, Nantes, France.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, 194:28–61.

Eduard H. Hovy, Roberto Navigli, and Simone Paolo Ponzetto. 2013. Collaboratively built semi-structured content and Artificial Intelligence: The story so far. *Artificial Intelligence*, 194:2–27.

Nancy Ide and Jean Véronis. 1993. Extracting knowledge bases from machine-readable dictionaries: Have we wasted our time? In *Proceedings of the Workshop on Knowledge Bases and Knowledge Structures*, pages 257–266, Tokyo, Japan.

Dan Klein and Christopher D. Manning. 2003. Fast Exact Inference with a Factored Model for Natural Language Parsing. In *Advances in Neural Information Processing Systems 15 (NIPS)*, pages 3–10, Vancouver, British Columbia, Canada.

Zornitsa Kozareva and Eduard H. Hovy. 2010. A Semi-Supervised Method to Learn and Construct Taxonomies Using the Web. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '10)*, pages 1110–1118, Seattle, WA, USA.

Olena Medelyan, David Milne, Catherine Legg, and Ian H. Witten. 2009. Mining meaning from Wikipedia. *International Journal of Human-Computer Studies*, 67(9):716–754.

Shachar Mirkin, Ido Dagan, and Eyal Shnarch. 2009. Evaluating the inferential utility of lexical-semantic resources. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 558–566, Athens, Greece.

Tom Mitchell. 2005. Reading the Web: A Breakthrough Goal for AI. *AI Magazine*.

Vivi Nastase and Michael Strube. 2013. Transforming Wikipedia into a large scale multilingual concept network. *Artificial Intelligence*, 194:62–85.

Vivi Nastase, Michael Strube, Benjamin Boerschinger, Caecilia Zirn, and Anas Elghafari. 2010. WikiNet: A Very Large Scale Multi-Lingual Concept Network. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Roberto Navigli and Paola Velardi. 2010. Learning Word-Class Lattices for Definition and Hypernym Extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 1318–1327, Uppsala, Sweden, July. Association for Computational Linguistics.

Patrick Pantel and Deepak Ravichandran. 2004. Automatically labeling semantic classes. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT 2013), Boston, Massachusetts, 2–7 May 2004*, pages 321–328.

Simone Paolo Ponzetto and Michael Strube. 2007. Deriving a large scale taxonomy from Wikipedia. In *Proceedings of the 22nd Conference on the Advancement of Artificial Intelligence (AAAI '07), Vancouver, B.C., Canada, 22–26 July 2007*, pages 1440–1445.

Simone Paolo Ponzetto and Michael Strube. 2011. Taxonomy induction based on a collaboratively built knowledge repository. *Artificial Intelligence*, 175(9-10):1737–1756.

Hoifung Poon, Janara Christensen, Pedro Domingos, Oren Etzioni, Raphael Hoffmann, Chloe Kiddon, Thomas Lin, Xiao Ling, Mausam, Alan Ritter, Stefan Schoenmackers, Stephen Soderland, Dan Weld, Fei Wu, and Congle Zhang. 2010. Machine Reading at the University of Washington. In *Proceedings of the 1st International Workshop on Formalisms and Methodology for Learning by Reading in conjunction with NAACL-HLT 2010*, pages 87–95, Los Angeles, California, USA.

Maria Ruiz-Casado, Enrique Alfonseca, and Pablo Castells. 2005. Automatic assignment of Wikipedia encyclopedic entries to WordNet synsets. In *Advances in Web Intelligence*, volume 3528 of *Lecture Notes in Computer Science*, pages 380–386. Springer Verlag.

Amit Singhal. 2012. Introducing the Knowledge Graph: Things, Not Strings. Technical report, Official Blog (of Google). Retrieved May 18, 2012.

Rion Snow, Dan Jurafsky, and Andrew Ng. 2006. Semantic taxonomy induction from heterogeneous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006)*, pages 801–808.

Fabian Suchanek and Gerhard Weikum. 2013. Knowledge harvesting from text and Web sources. In *IEEE 29th International Conference on Data Engineering (ICDE 2013)*, pages 1250–1253, Brisbane, Australia. IEEE Computer Society.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2008. YAGO: A large ontology from Wikipedia and WordNet. *Journal of Web Semantics*, 6(3):203–217.

Daniele Vannella, David Jurgens, Daniele Scarfini, Domenico Toscani, and Roberto Navigli. 2014. Validating and Extending Semantic Knowledge Bases using Video Games with a Purpose. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, Baltimore, USA.

Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. OntoLearn Reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39(3):665–707.