

# Taxonomic Data Integration from Multilingual Wikipedia Editions

Gerard de Melo · Gerhard Weikum

Received: date / Accepted: date

**Abstract** Information systems are increasingly making use of taxonomic knowledge about words and entities. A taxonomic knowledge base may reveal that the Lago di Garda is a lake, and that lakes as well as ponds, reservoirs, and marshes are all bodies of water. As the number of available taxonomic knowledge sources grows, there is a need for techniques to integrate such data into combined, unified taxonomies. In particular, the Wikipedia encyclopedia has been used by a number of projects, but its multilingual nature has largely been neglected. This paper investigates how entities from all editions of Wikipedia as well as WordNet can be integrated into a single coherent taxonomic class hierarchy. We rely on linking heuristics to discover potential taxonomic relationships, graph partitioning to form consistent equivalence classes of entities, and a Markov chain-based ranking approach to construct the final taxonomy. This results in MENTA (Multilingual Entity Taxonomy), a resource that describes 5.4 million entities and is one of the largest multilingual lexical knowledge bases currently available.

**Keywords** Taxonomy induction · Multilingual · Graph · Ranking

## 1 Introduction

*Motivation.* Capturing knowledge in the form of machine-readable semantic knowledge bases has been a long-standing goal in computer science, information science, and knowledge management. Such resources have facilitated tasks like query expansion [34], semantic search [48], faceted search [8], question answering [69], semantic document clustering [19], clinical decision support systems [14], and many more. Knowledge about taxonomic relationships is particularly important. A taxonomic knowledge base may reveal that the Lago di Garda is a lake, and that lakes as well as ponds, reservoirs, and marshes are all bodies of water.

As the Web matures, more and more sources of taxonomic knowledge are appearing, and there is an increasing need for methods that combine individual

---

Max Planck Institute for Informatics  
Saarbrücken, Germany  
E-mail: {gdemelo,weikum}@mpi-inf.mpg.de

taxonomic data into unified taxonomic knowledge bases. While there has been research on extracting and creating individual taxonomies and on finding alignments between two taxonomies, little attention has been paid to this new challenge of *taxonomic data integration*, which entails merging taxonomic information from several possibly noisy data sources.

We focus in particular on Wikipedia, the open, community-developed online encyclopedia that in the past few years has been recognized as a valuable source of taxonomic knowledge. Projects like DBpedia [6], YAGO [76], Intelligence-in-Wikipedia [84], and Freebase (*freebase.com*) have exploited the semi-structured nature of Wikipedia to produce valuable repositories of formal knowledge that are orders of magnitude larger than hand-crafted resources like SUMO [54], OpenCyc (*open-cyc.org*), or WordNet [31]. To date, however, these extraction efforts have largely neglected the significant potential of Wikipedia’s multilingual nature. While DBpedia and some other knowledge bases do extract abstracts and other information also from non-English versions, such information is only fully integrated with the English knowledge when a given article has a corresponding article in the English Wikipedia. Certainly, the English Wikipedia is by far the most comprehensive version. Yet, its articles make up only 25% among those of the 20 largest Wikipedias<sup>1</sup>.

Although it is certainly possible to construct separate taxonomies from different language-specific editions of Wikipedia, an algorithm that is able to aggregate and combine information from each of these data sources is able to produce a cleaner output taxonomy while still retaining most of the data source-specific information.

The algorithm we propose considers the interdependencies between many data sources, in our case over 200 different editions of Wikipedia as well as WordNet. It connects the noisy and sometimes conflicting evidence that these sources provide and derives a single unified taxonomy that is experimentally shown to have a higher quality than the initial inputs.

We use the algorithm to construct MENTA – Multilingual Entity Taxonomy – a large-scale taxonomic knowledge base that covers a significantly greater range of entities than existing knowledge bases. Additionally, MENTA enables tasks like semantic search also in languages other than English, for which existing taxonomies are often very limited or entirely non-existent. Finally, we also hope that MENTA will facilitate decidedly multilingual applications like cross-lingual information retrieval [29, 9], machine translation [42], or learning transliterations [63].

*Problem Statement.* As input we have a set of knowledge sources. The entity identifiers used in these knowledge sources need to be connected using some heuristics. This results in a large but incomplete set of unreliable, weighted statements linking entities to parent entities (taxonomic links) or to equivalent entities (**equals** arcs). For a given entity, we often have many candidate parents from different sources with different weights, and different parents may or may not be related to each other in terms of **equals** and **subclass** arcs (see Figure 2 for an example scenario).

The aim is to aggregate these unreliable, incomplete taxonomic links between entities from different sources into a single more reliable and coherent taxonomy.

<sup>1</sup> Computed by dividing the number of articles in the English Wikipedia by the sum of all articles in the 20 largest Wikipedias with respect to the number of articles, as of April 2012.

The output should be a clean, reliable knowledge base where the entities share a single upper-level core rather than having a diverse range of separate taxonomies. Schematically, this task is depicted in Figure 1.

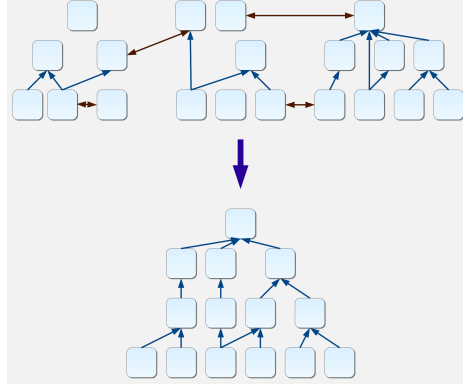


Fig. 1: Taxonomic integration strategy

*Contribution.* Our contributions include the following.

1. **Link Prediction Heuristics:** We show information can be extracted from many multilingual versions of Wikipedia, and devise heuristics to interconnect articles, categories, infoboxes, as well as WordNet senses in multiple languages. The links capture equivalence between items as well as taxonomic relationships.
2. **Taxonomic Data Integration:** Algorithmically, we show how one can aggregate and integrate the individual original links between entities, which are somewhat unreliable, into a single more reliable and coherent taxonomy, as sketched in Figure 1. This algorithm has two key features:
  - It produces aggregated rankings of taxonomic parents for a given entity based on the amount of evidence for each candidate parent.
  - The rankings take into account dependencies between possible parents. In particular, multiple parents might be equivalent or might stand in some sort of hierarchical relation to one another. In such cases, computing a ranking is a lot less straightforward.
 Our algorithm addresses this challenge by deriving the output for a specific entity using the stationary distribution of a Markov chain, in the spirit of PageRank, but adapted to our specific setting. After a final filtering step, we obtain a coherent taxonomic knowledge base.
3. **The MENTA Knowledge Base:** We show how the framework can be applied to produce a large-scale resource called MENTA, which is novel in several respects.
  - **Extended Coverage of Entities:** The taxonomy draws on all existing editions of Wikipedia and hence includes large numbers of local places, people, products, etc. not covered by the English Wikipedia. For example, the Quechua Wikipedia has an article about the Bolivian salt lake Salar de Coipasa, and the Japanese one has an article about Italian Parma ham.

- **Ranked Class Information:** Individual entities are linked via **instance** statements to classes (e.g. **City**, **Airline company**, etc.) by exploiting complementary clues from different Wikipedia languages. The output is ranked, because it is useful to capture e.g. that the Colorado River being a river is more salient than it being a border of Arizona.
- **Coherent Taxonomy:** While Wikipedia is an excellent source of semi-structured knowledge, it lacks an ontologically organized taxonomy. The category systems of Wikipedia i) fail to distinguish classes from topic labels (Yellowstone National Park is a natural park but not a ‘*History of Wyoming*’) ii) tend to lack a clear organization especially at the most abstract level, and iii) differ substantially between different languages. MENTA’s clean, global hierarchical organization connects all entities in the knowledge base, even if they originate from different editions of Wikipedia or from WordNet.

With these features, MENTA goes beyond other repositories of semantic knowledge. For instance, DBpedia and YAGO do not have a multilingual upper-level ontology. None of the existing taxonomies have managed to accommodate culture-specific entities from non-English Wikipedia editions. Even for those entities that are covered, the DBpedia Ontology provides class information only for around a third. Likewise, in the field of multilingual taxonomies or hierarchically-organized multilingual lexical knowledge bases, our knowledge base surpasses all existing resources in the number of entities described. The largest comparable resources, BabelNet [52] and WikiNet [51], were developed in parallel to MENTA and also draw on multilingual information from Wikipedia. However, they do not exploit all 270+ Wikipedia editions and do not emphasize producing a coherent taxonomy. MENTA is freely available under an open-source license<sup>2</sup>.

*Overview.* Section 2 lays out how information is extracted from Wikipedia and represented in a form amenable to further processing. Section 3 then introduces the heuristics that are used to interlink entities and provide the input for the taxonomy induction step. Section 4 describes the actual algorithm for producing the unified knowledge base with a single taxonomic class hierarchy. Section 5 describes the system architecture and online interface. Section 6 evaluates the algorithm and the resulting knowledge base. Section 7 describes related knowledge bases and approaches. Finally, Section 8 provides concluding remarks.

## 2 Knowledge Extraction

### 2.1 Representation Model

We regard taxonomies as knowledge bases that describe relationships between entities. Entities are taken to include both individual entities as well classes. A taxonomy of this form could describe the **Lago di Garda** as an *instance* of a **Subalpine lake**, **Subalpine lake** as a *subclass* of **Lake**, **Lake** as a subclass of **Body of Water**, and so on, up to a universal root node, often called **Entity**.

<sup>2</sup> <http://www.mpii.de/yago-naga/menta/>

**Definition 1** A *statement* is a weighted labelled arc in  $V \times V \times \Sigma \times \mathbb{R}_0^+$ , where  $V$  is a universal set of entity identifiers (nodes) and  $\Sigma$  is a set of arc labels (relations). A statement  $(x, y, r, w)$  expresses that two entities  $x, y$  stand in relation  $r$  to each other with weight  $w$ , where a weight of 0 means there is no evidence, and strictly positive values quantify the degree of confidence in the statement being true.

**Definition 2** A *knowledge base*  $K$  is a weighted labelled multi-digraph  $G = (V, A, \Sigma)$  where:  $V$  set of entity identifiers that constitute the nodes in the graph,  $A$  is a set of statements as defined above, and  $\Sigma$  is the set of arc labels.

In our case,  $V$  contains entity identifiers for Wikipedia pages (including categories and infobox templates), word senses (“synsets”) defined by WordNet [31], as well as string literals with language designators. The arc label set  $\Sigma$  includes:

- **equals**: identity of entities (i.e. two entity identifiers refer to the same entity)
- **subclass**: the relation between a class and a subsuming generalization of it, i.e. a parent class
- **instance**: the relation between an individual entity and a class it is an instance of (its class, type, or role)
- **means**: the meaning relationship between a language-specific string entity (a word or a name) and another entity

A statement might express an **instance** relationship between **University of Trento** and **University** with confidence 1, or a **means** relationship between the Polish name ‘*Trydent*’ and the city of Trento. Such statements can easily be cast into an RDF [37] form, if reification is used to capture the confidence values.

## 2.2 Extraction from Wikipedia

*Entities.* Before aggregating information, we parse the raw XML and wiki-markup-based Wikipedia dumps, extract relevant information, and cast it into our representation model to facilitate further processing. In particular, each article page (including redirect pages), category page, or template page in an edition of Wikipedia is given a preliminary entity identifier. Unfortunately, not all information necessary for parsing the Wikipedia dumps is available from within the dumps alone. We additionally query the web services provided by each server to find out for instance that in the Tagalog Wikipedia, titles starting with “Kategorya:” refer to categories (in addition to the default “Kaurian:” and the English “Category:”, which are also accepted). Such information is normalized, so as to obtain canonical entity identifiers. Being able to recognize categories is also helpful at a later stage when constructing the taxonomy.

*Statements.* Additional information about entities and meta-data about articles that may be of use later on is extracted and stored with appropriate relations. In particular, we capture template invocations, cross-lingual “interwiki” links, redirects, multimedia links, category links, and optional factual statements (**locatedIn**, **bornIn**, and so on).

Additionally, we create short description glosses for each article entity (**hasGloss**) by processing wikitext and HTML mark-up and attempting to identify

the first proper paragraph in an article’s wikitext mark-up (skipping infoboxes, pictures, links to disambiguation pages, etc.). If this first paragraph is too long, i.e. the length is greater than some  $l$ , a sentence boundary is identified in the vicinity of the position  $l$ .

*Meanings.* Article titles allow us to create **means** statements connecting entities with language-specific strings (labels or names) that refer to them. The original article title is modified by removing any additional qualifications in parentheses, e.g. ‘*School (discipline)*’ becomes ‘*School*’. Some articles use special markup to provide the true capitalization, e.g. ‘*iPod*’ instead of ‘*IPod*’. If no markup is provided, we check for the most frequent capitalization variant within the article text.

### 3 Linking Functions

Given our goal of creating a single more coherent knowledge base from multiple data sources, especially from the different editions of Wikipedia and WordNet, our strategy will be to first expose possible connections between different nodes using several heuristics. After that, in a second step described later on in Section 4, we integrate these noisy inputs to induce a shared taxonomy.

For the first step, we rely on so-called linking functions to identify how different entities relate to each other. In particular, Section 3.1 introduces **equals** linking functions that identify identical entities, and Sections 3.2 and 3.3 present linking functions for the **subclass** and **instance** relations.

**Definition 3** A *linking function*  $l_r : V \times V \rightarrow \mathbb{R}_0^+$  for a specific relation  $r \in \Sigma$  is a function that yields confidence weight scores  $l_r(x, y) \in \mathbb{R}_0^+$  and is used to produce statements  $(x, y, r, l_r(x, y))$  for pairs of entity identifiers  $x, y$ .

Given a set of **equals** linking functions  $L_e$ , a set of **subclass** linking functions  $L_s$ , and a set of **instance** linking functions  $L_i$ , Algorithm 3.1 shows how the input graph is extended with appropriate links. For each linking function  $l \in L_e \cup L_s \cup L_i$ , we additionally assume we have a candidate selection function  $\sigma_l$ , which for a given node  $x \in V$  yields a set  $\sigma_l(x) \subseteq V$  containing all nodes  $y$  that are likely to have non-zero scores  $l(x, y) > 0$ .

---

#### Algorithm 3.1 Linking function application

---

```

1: procedure CREATELINKS( $G_0 = (V_0, A_0, \Sigma)$ ,  $L_e, L_s, L_i, \{\sigma_l \mid l \in L_e \cup L_s \cup L_i\}$ )
2:   for all  $l$  in  $L_e$  do                                     ▷ for each equals linking function
3:      $A_0 \leftarrow A_0 \cup \{(x, y, r, w) \mid x \in V_0, y \in \sigma_l(x), r = \text{equals}, w = l(x, y)\}$ 
4:   for all  $l$  in  $L_s$  do                                     ▷ for each subclass linking function
5:      $A_0 \leftarrow A_0 \cup \{(x, y, r, w) \mid x \in V_0, y \in \sigma_l(x), r = \text{subclass}, w = l(x, y)\}$ 
6:   for all  $l$  in  $L_i$  do                                     ▷ for each instance linking function
7:      $A_0 \leftarrow A_0 \cup \{(x, y, r, w) \mid x \in V_0, y \in \sigma_l(x), r = \text{instance}, w = l(x, y)\}$ 
8:   return  $G_0 = (V_0, A_0, \Sigma)$ 

```

---

Later on, we will explain how the output of somewhat unreliable linking functions can be aggregated to provide meaningful results. Which heuristics are appropriate for a given input scenario depends on the knowledge sources involved. We

will now describe the specific choices of linking functions that we use to connect entities in different language-specific editions of Wikipedia as well as WordNet.

### 3.1 Equality Link Heuristics

We use the following linking functions to generate **equals** arcs between two entity identifiers  $x, y$ .

#### 3.1.1 Cross-Lingual Linking

If there is a cross-lingual interwiki link from  $x$  to  $y$  in Wikipedia, e.g. from **Trydent** in the Polish Wikipedia to **Trento** in the English one, this function yields 1, otherwise 0.

#### 3.1.2 Category-Article Linking

The category-article linking function returns 1 when  $x, y$  correspond to a category and an article, respectively, known to be about the same concept, e.g. the category **Abugida writing systems** and the article **Abugida**. This is detected by checking for specific template invocations on the category page.

#### 3.1.3 Supervised WordNet Disambiguation

A Wikipedia entity like **Degree (school)** could match several different WordNet entities for the word ‘degree’, e.g. degree as a position on a scale, or as the highest power of a polynomial. Likewise, Wikipedia provides several different candidates for each WordNet entity, e.g. degree as the number of edges incident to a vertex of a graph, or ‘Degree’ as a brand name. In order to reliably assess the similarity between a Wikipedia article, category, or infobox and a WordNet synset, we relied on a supervised linking function to disambiguate possible meanings. The function relies on Ridge Regression [11] to derive a model from a small set of manually labelled training examples (cf. Section 6.2.1). It uses three major signals as features.

*Term Overlap.* The term overlap feature quantifies the degree of similarity between the respective human language terms associated with entities. Here, the set  $\text{terms}(x)$  for a Wikipedia entity  $x$  is given by its title (after removing additional qualifications and detecting the correct capitalization, as mentioned earlier) and titles of its redirection articles. A set of terms for a WordNet entity is retrieved from the English, Arabic [68], Catalan [10], Estonian [59], Hebrew [60], and Spanish [4] wordnets as well as from MLSN [21].

For a Wikipedia entity  $x$  and a WordNet entity  $y$ , the term overlap feature is then computed as:

$$\sum_{t_x \in \text{terms}(x)} \max_{t_y \in \text{terms}(y)} \phi_x(t_x, x) \phi_y(t_y, y) \text{sim}(t_x, t_y) \quad (1)$$

Here,  $\text{sim}(t_x, t_y)$  is a simple similarity measure between terms that returns 1 if the languages match and the strings are equal after lemmatizing, and 0 otherwise.

For Wikipedia, the additional term weighting  $\phi_x$  generally yields 1, while for WordNet multiple different versions of  $\phi_y$  are used in separate features. One option is to have  $\phi_y$  return  $1/n$  when  $n$  different meanings of  $t_y$  are listed in WordNet. Another option is to use

$$\phi_y(t_y, y) = \frac{1}{\text{rank}(t_y, y) + \frac{1}{2}}$$

where  $\text{rank}(t_y, y)$  is the rank of a synset  $y$  for a word  $t_y$  as delivered by WordNet (e.g. 1, 2, ...). WordNet places more important word senses first. Finally,

$$\phi_y(t_y, y) = \frac{\text{freq}(t_y, y)}{\sum_{\substack{y' \text{ is a noun} \\ \text{synset for } t_y}} \text{freq}(t_y, y')}$$

is used as well, where  $\text{freq}(t_y, y)$  provides the frequency of word  $t_y$  with sense  $y$  in the sense-annotated SemCor corpus.

It turns out that determining the right capitalization of terms aids in avoiding incorrect matches. WordNet synsets for ‘*college*’ will then only match articles about colleges but not articles about films or subway stops called ‘*College*’.

*Cosine Similarity.* The cosine vector similarity feature is computed as  $\mathbf{v}_x^T \mathbf{v}_y / (\|\mathbf{v}_x\|_2 \|\mathbf{v}_y\|_2)^{-1}$  for vectors  $\mathbf{v}_x, \mathbf{v}_y$  derived for the short description gloss extracted from the English Wikipedia in Section 2.2 and the gloss and related terms provided by WordNet, respectively. The vectors are created using TF-IDF scores after stemming using Porter’s method.

*Primary Sense Heuristic.* This feature is computed by taking the set of unqualified English titles for the Wikipedia entity  $x$  or any of its redirects, and then counting for how many of them the WordNet synset  $y$  is listed as the primary noun sense in WordNet. A Wikipedia title like ‘*College*’ is considered unqualified if it does not include an additional qualification in parentheses, unlike ‘*College (canon law)*’. The most frequent sense of ‘*college*’ listed in WordNet is much more likely to correspond to Wikipedia’s ‘*College*’ article than to pages with additional qualifications like ‘*College (canon law)*’ or ‘*College (1927 film)*’. Unqualified titles reflect the most important meaning of words as chosen by Wikipedia editors, and thus are more likely to correspond to the first sense of those words listed in WordNet.

*Supervised Model.* Together, these three signals allow us to learn a regression model that assesses whether a Wikipedia article and a WordNet synset are likely to match. The contribution of the three signals may vary from case to case. For some articles, we may find an English redirect title that matches a WordNet synset as well a Spanish title that matches the equivalent synset in the Spanish WordNet. In other cases, only the English title may match but we might additionally see that this English title’s first sense is precisely the WordNet synset we are considering and that additionally the WordNet synset’s gloss has a high cosine similarity with the gloss extracted from the article. Note that these rich signals provide more information than is normally available when performing conventional word sense disambiguation for words occurring in a text.



### 3.1.4 Redirect Matching

Many projects treat redirect titles in Wikipedia as simple alias names of an entity. However, the meanings of many redirect titles differ significantly from those of their respective redirect target pages. For instance, there are redirects from **Physicist** (i.e. human beings) to **Physics** (a branch of science) and from **God does not play dice** to **Albert Einstein**. Large numbers of redirects exist from song names to album names or artist names, and so on. We decided to conservatively equate redirects with their targets only in the following two cases.

1. The titles of redirect source and redirect target match after parenthesized substring removal, Unicode NFKD normalization [22], diacritics and punctuation removal, and lower-case conversion. Hence **London** would match **London (England)** and **LONDON**, but not **Southwest London** or **Climate of London**.
2. The redirect uses certain templates or categories that explicitly indicate co-reference with the target (alternative names, abbreviations, etc.).

Other redirects still have a chance of being connected to their targets later on, by the methods described in Section 4.1.

### 3.1.5 Infobox Matching

This linking function returns a constant  $w > 0$  when an infobox template like **Infobox university** is matched with an article or category having a corresponding title, in this case **University**, and 0.0 otherwise. We chose  $w = 0.5$  because these mappings are not as reliable as interwiki links or redirect links. The function does not consider article titles with additional qualifications as matching, so **University (album)** would not be considered.

## 3.2 Subclass Link Heuristics

Subclass linking functions use simple heuristics to connect a class  $x$  to its potential parent classes  $y$ .

### 3.2.1 Parent Categories

The parent category linking function checks if entities  $x$  for Wikipedia categories can be considered subclasses in the ontological sense of entities  $y$  for their own parent categories as listed in Wikipedia.

To accomplish this, it ensures that both  $x$  and  $y$  are likely to be categories denoting genuine classes. A genuine class like **Universities** can have instances as its class members (individual universities, ontologically speaking, are regarded as instances of **Universities**). In contrast, other categories like **Education** or **Science education** merely serve as topic labels. It would be wrong to say that the University of Trento “is an” **Education**. For distinguishing the two cases automatically, we found that the following heuristic generalizes the singular/plural heuristic proposed for YAGO [76] to the multilingual case:

- headword nouns that are countable (can have a plural form) tend to indicate genuine classes

- headword nouns that are uncountable (exist only in singular form) tend to be topic tags

Hence, we take the titles of a category as well as its cross-lingual counterparts, remove qualifications in parentheses, and, if available, rely on a parser to retain only the main headword. In practice, we exclusively use the English Link Grammar parser [71]. For large numbers of non-English categories, it suffices to work with the entire string after removing qualifications, e.g. the German Wikipedia uses titles like **Hochschullehrer (Berlin)** rather than titles like **German academics**. In most other cases, the Markov Chain Taxonomy Induction algorithm will succeed at ensuring that taxonomic links are nevertheless induced. We then check that whatever term remains is given in plural (for English), or is countable (in the general case). Countability information is extracted from WordNet and Wiktionary (*wiktionary.org*), the latter using regular expressions. We also added a small list of Wikipedia-specific exceptions (words like ‘*articles*’, ‘*stubs*’) that are excluded from consideration as classes.

The linking function returns 1 if  $y$  is a parent category of  $x$  and both  $x$  and  $y$  are likely to be genuine classes, and 0 otherwise.

### 3.2.2 Category-WordNet Subclass Relationships

If  $x$  is a category, then the headword of its title also provides a clue as to what parent classes are likely in the input wordnets. For instance, a category like **University book publishers** has ‘*publishers*’ as a headword. While we need the headword to be covered by the input wordnets, it suffices to use the English WordNet and perhaps a few other ones. As we will later see, even if one were to use only Princeton WordNet, the Markov Chain Taxonomy Induction algorithm could easily integrate most categories, because the majority of non-English categories will have **equals** arcs to English categories or **subclass** links ultimately leading to an article or category that is connected to WordNet.

We again relied on supervised learning to disambiguate possible meanings of a word, as earlier employing Ridge Regression [11] to learn a model that recognizes likely entities based on a labelled training set (see Section 6.2.2). The main features are again of the form

$$\sum_{t_x \in \text{terms}(x)} \max_{t_y \in \text{terms}(y)} \phi_x(t_x, x) \phi_y(t_y, y) \text{sim}_{\text{hw}}(t_x, t_y) \quad (2)$$

This is similar to Equation 1, however  $\text{sim}_{\text{hw}}(t_x, t_y)$  matches with headwords of titles  $t_x$  rather than full titles  $t_x$  if such information is available. As for the **subclass** links, qualifications in parentheses are removed, and then the Link Grammar parser is used to retain only the headword [71] if possible. Additionally,  $\phi_x(t_x, x)$  will be 1 if  $t_x$  is in plural or countable and 0 otherwise, allowing us to distinguish topic labels from genuine classes. The second weighting function  $\phi_y(t_y, b)$  again uses the number of alternative meanings as well as synset rank and corpus frequency information. Apart from this, the linking also relies on the cosine similarity feature used earlier for **equals**. Together, these features allow the model to disambiguate between relevant WordNet synsets. A few exceptions are specified manually, e.g. ‘*capital*’, ‘*single*’, ‘*physics*’, ‘*arts*’, and Wikipedia-specific ones like ‘*articles*’, ‘*pages*’, ‘*templates*’.

### 3.2.3 WordNet Hypernymy

WordNet’s notion of hypernymy between synsets is closely related to the **subclass** relation. The WordNet hypernymy linking function hence returns 1 if  $y$  is a hypernym of  $x$  in WordNet, and 0 otherwise.

## 3.3 Instance Link Heuristics

Instance linking functions link individual entities to their classes.

### 3.3.1 Infoboxes

A **University** infobox placed in a Wikipedia article is a very strong indicator of the article being about a university. The instance linking function returns a constant  $w_{\text{infobox}} > 0$  if  $y$  is recognized as an infobox template that occurred on the page of the article associated with  $x$ , and 0 otherwise. Since infoboxes are incorporated into Wikipedia articles by means of simple template invocations, heuristics need to be used to distinguish them from other sorts of template invocations. For this, we rely on a list of suffixes and prefixes (like “Infobox”) for different languages. The **instance** links generated by this linking function are useful later on, because we will also have **equals** links between infobox templates and articles, as described in Section 3.1.

### 3.3.2 Categories

Entities for articles like **Free University of Bozen-Bolzano** are made instances of certain categories, e.g. **Universities in Italy**, but not of topic categories like **Bolzano**. If  $y$  is a Wikipedia category for the article associated with  $x$ , the category linking function assesses whether a headword of  $y$  (or of its interwiki translations) is in plural or countable, and returns 1 if this is the case, and 0 otherwise, as earlier for subclass relations.

We will now explain what these linking functions give us and what needs to be done in order to obtain a more coherent output knowledge base.

## 4 Taxonomy Induction

Applying the linking functions to the input as in Algorithm 3.1, we obtain a graph  $G_0 = (V_0, A_0, \Sigma)$  with an extended arc set  $A_0$  connecting entities from multiple knowledge sources to each other, in our case articles, categories, infoboxes (from different editions of Wikipedia), as well as WordNet entities. As shown in Figure 2, the connections include **equals** statements (red bidirectional arrows) representing equivalence, **subclass** statements connecting categories and WordNet entities to parent classes, and **instance** statements connecting articles to categories and infoboxes (both depicted as blue unidirectional arrows).

However, due to the noisy heuristic nature of these arcs and the fact that these entities come from different sources, it is not trivial to recognize that ‘*Fersental*’

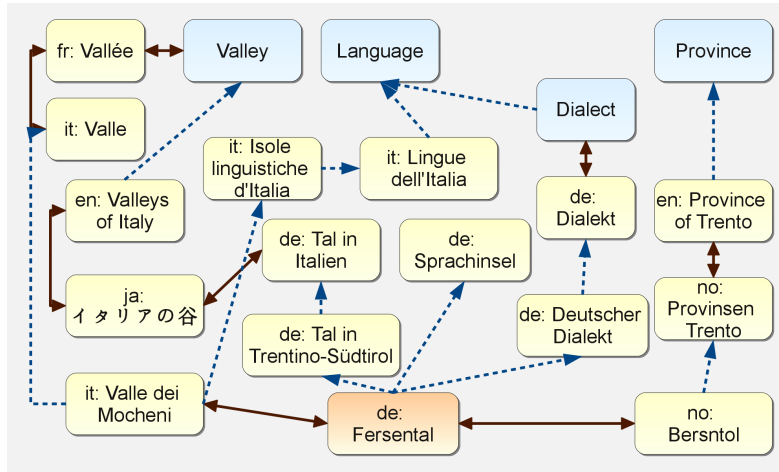


Fig. 2: Simplified illustration of noisy input from heuristics, including **equals** (red bidirectional arrows) and **instance/subclass** links (blue unidirectional arrows)

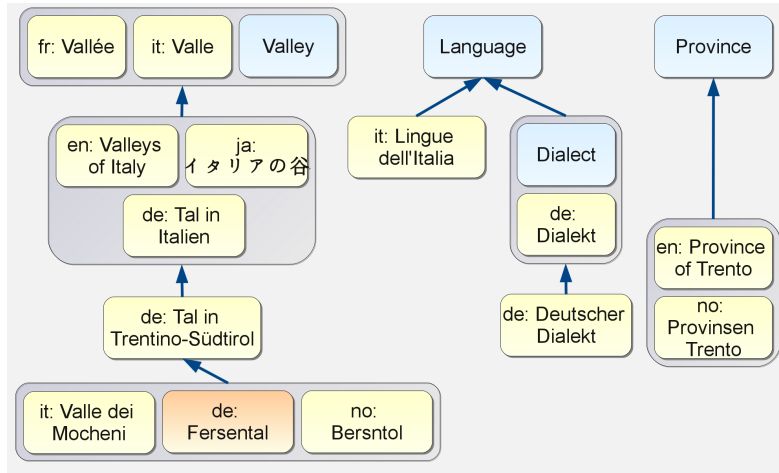


Fig. 3: Relevant sample of the desired output

is a valley rather than a language. In fact, in reality, we may have more than 50 languages and many more potential parents for an entity. What is needed is a way to aggregate information and produce the final, much cleaner and more coherent knowledge base, which would ideally include what is depicted in Figure 3. We proceed in three steps. The first step aggregates entity identifiers referring to the same entity by producing consistent equivalence classes. In the second step, taxonomic information from different linking functions is aggregated to produce a clean taxonomy. A final step filters this taxonomy to make it even more consistent.

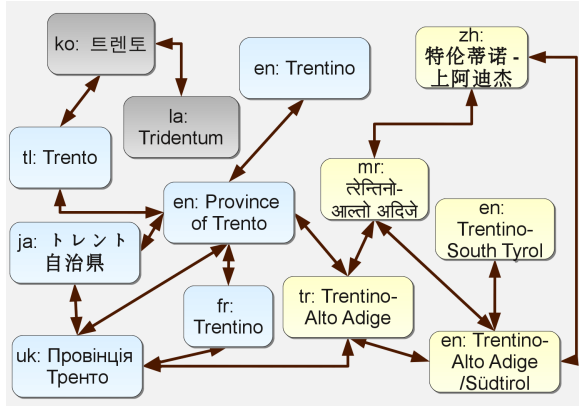


Fig. 4: Connected component with inaccurate links (simplified)

#### 4.1 Consistency of Equivalence Information

There will often be multiple entity identifiers that refer to the same entity and that are connected by **equals** statements. For instance, the German **Fersental** is equivalent to the corresponding Italian, Norwegian, and other articles about the valley. It will sometimes be convenient to jointly refer to all of these equivalents.

To make the knowledge base more coherent, one key ingredient is taking into account the symmetry and transitivity of equivalence. In practice, we may have an infobox in some non-English edition with an **equals** arc to an article, which has an **equals** arc to a category, which in turn has an interwiki link to an English category, and so on.

This leads us to the following definition to capture the weakly connected components corresponding to the symmetric, transitive closure of **equals**.

**Definition 4** (e-component) In a knowledge base  $G = (V, A, \Sigma)$ , an *e-component*  $E \subseteq V$  for some entity  $v_0 \in V$  is a minimal set of entities containing  $v_0$  such that  $v \in E$  for all  $u \in E$ ,  $v \in V$  with statements  $(u, v, r, w) \in A$  or  $(v, u, r, w) \in A$  (with  $r = \text{equals}$ ,  $w > 0$ ). We use the notation  $E(v_0)$  to denote the e-component containing a node  $v_0$ .

Due to the heuristic nature of the equality linking functions, it often occurs that two entities  $u, v$  are transitively identified within an e-component, although we are quite sure that they should not be. For instance, we may have two different Wikipedia articles linked to the same WordNet synset. In some cases, the input from Wikipedia is imprecise, e.g. the Catalan article about the city of Bali in Rajasthan, India, as of February 2011, is linked to the Hindi article about the Indonesian island of Bali.

Figure 4 shows a connected component that conflates multiple different entities. The Latin Wikipedia’s **Tridentum** refers to the city of Trento, which is distinct from the larger Trentino (also known as the Province of Trento). Finally, Trentino-Alto Adige/Südtirol refers to an even larger autonomous region that includes not only Trentino but also South Tyrol (also known as Alto Adige, or the Province of Bolzano-Bozen). In the figure, we see that these entities are not clearly sep-

arated. Among other things, we would like to state that **Trentino-South Tyrol** and **Trentino-Alto Adige/Südtirol** are distinct from **Trentino** and **Province of Trento** (perhaps remaining agnostic about whether **Trentino** and **Province of Trento** are distinct or not). In general, we may have several sets of entities  $D_{i,1}, \dots, D_{i,l_i}$ , for which we assume that any two entities  $u, v$  from different sets are pairwise distinct with some degree of confidence or weight. In our example,  $D_{i,1} = \{\text{Trentino-South Tyrol}, \text{Trentino-Alto Adige/Südtirol}\}$  would be one set, and  $D_{i,2} = \{\text{Trentino}, \text{Province of Trento}\}$  would be another set. Formally, we can make the following definition.

**Definition 5** (Distinctness Assertions) Given a set of nodes  $V$ , a *distinctness assertion* is a collection  $D_i = (D_{i,1}, \dots, D_{i,l_i})$  of pairwise disjoint (i.e.  $D_{i,j} \cap D_{i,k} = \emptyset$  for  $j \neq k$ ) subsets  $D_{i,j} \subset V$  that expresses that any two nodes  $u \in D_{i,j}, v \in D_{i,k}$  from different subsets ( $j \neq k$ ) are asserted to be distinct from each other with some weight  $w(D_i) \in \mathbb{R}$ .

We found that many inconsistent e-components can be identified automatically with the following distinctness assertions. Among other things, they discourage articles from the same Wikipedia from being merged, multiple WordNet synsets from being merged, and disambiguation pages from being mixed up with regular articles.

**Criterion 1** (*Distinctness between articles from the same Wikipedia edition*) For each language-specific edition of Wikipedia, a separate assertion  $(D_{i,1}, D_{i,2}, \dots)$  can be made, where each  $D_{i,j}$  contains an individual *article* together with its respective redirection pages. Two articles from the same Wikipedia very likely describe distinct concepts unless they are redirects of each other. For example, ‘*Georgia (country)*’ is distinct from ‘*Georgia (U.S. State)*’. Additionally, there are also redirects that are clearly marked by a category or template as involving topic drift, e.g. redirects from songs to albums or artists, from products to companies, etc. We keep such redirects in a  $D_{i,j}$  distinct from the one of their redirect targets.

**Criterion 2** (*Distinctness between categories from the same Wikipedia edition*) For each language-specific edition of Wikipedia, a separate assertion  $(D_{i,1}, D_{i,2}, \dots)$  is made, where each  $D_{i,j}$  contains a *category* page together with any redirects. For instance, ‘*Category:Writers*’ is distinct from ‘*Category:Writing*’.

**Criterion 3** (*Distinctness for links with anchor identifiers*) The English ‘*Division by zero*’, for instance, links to the German ‘*Null#Division*’. The latter is only a part of a larger article about the number zero in general, so we can make a distinctness assertion to separate ‘*Division by zero*’ from ‘*Null*’. In general, for each interwiki link or redirection with an anchor identifier, we add an assertion  $(D_{i,1}, D_{i,2})$  where  $D_{i,1}, D_{i,2}$  represent the respective articles without anchor identifiers.

**Criterion 4** (*Distinctness of WordNet Synsets*) We assume that WordNet does not contain any duplicate synsets and add a distinctness assertion  $(D_{i,1}, D_{i,2}, \dots)$ , consisting of a singleton set  $D_{i,j} = \{v\}$  for each entity  $v$  from WordNet.

**Criterion 5** (*Distinctness from Disambiguation Pages*) We add an assertion  $(D_{i,1}, D_{i,2})$  where  $D_{i,1}$  contains all articles recognized as disambiguation pages, and  $D_{i,2}$  contains all articles not recognized as disambiguation pages. In Wikipedia, disambiguation pages are special pages that provide a list of available articles for ambiguous titles.

The criteria mentioned above are used to instantiate distinctness assertions for e-components. The assertion weights are tunable; the simplest choice is using a uniform weight for all assertions (note that these weights are different from the edge weights in the graph). We will revisit this issue in our experiments.

Note that we could also have chosen not to remain that faithful to WordNet and only enforce distinctness between different branches of entities within WordNet, e.g.  $(D_{i,1}, D_{i,2})$  where  $D_{i,1}$  contains all abstract entities in WordNet and  $D_{i,2}$  contains all physical entities in WordNet. Since we are aiming at a more precise upper-level ontology, we decided to maintain WordNet’s fine-grained sense distinctions.

*Algorithm.* To reconcile the **equals** arcs with the distinctness information, we often need to remove edges, or alternatively we may choose to ignore certain distinctness information. Separating two entities while removing a minimal (weighted) number of edges corresponds to computing minimal graph cuts. Unfortunately, we often have multiple pairs that simultaneously need to be separated, which is NP-hard and APX-hard. To cope with this, we first apply generic graph partitioning heuristics [24] to break up very large sparsely connected components into individual, much more densely connected clusters. On each of these densely connected clusters, we then apply a more accurate algorithm. We first solve a linear program using CPLEX, which gives us an optimal fractional solution, and then use a region growing technique that gives us a logarithmic approximation guarantee. See our prior work [46] for details. In a few cases, the LP solver may time out, in which case we resort to computing minimal  $s$ - $t$  cuts [27] between individual pairs of entities that should be separated. Minimal  $s$ - $t$  cuts can be computed efficiently in  $O(VE^2)$  or  $O(V^2E)$  time. The statements corresponding to the cut edges are removed, and hence we obtain small e-components that should no longer conflate different concepts.

In a few rare cases, the LP solver may time out even for small partitions, in which case we resort to computing minimal  $s$ - $t$  cuts [27] between individual pairs of entities that should be separated. Minimal  $s$ - $t$  cuts can be computed efficiently in  $O(VE^2)$  or  $O(V^2E)$  time. The statements corresponding to the cut edges are removed, and hence we obtain smaller e-components that should no longer conflate different concepts.

## 4.2 Aggregated Ranking

### 4.2.1 Requirements

Having made the **equals** arcs consistent, we then proceed to build the class hierarchy. In order to create the final output taxonomy, we will reconsider which entities to choose as superordinate taxonomic parents for a given entity. In doing so, the following considerations will need to be acknowledged.

First of all, the taxonomic arcs provided as inputs in general are not all equally reliable, as many of them originate from heuristic linking functions. The input arcs are equipped with statements weights that indicate how much we can trust them.

*Property 1 (Ranking)* The output should be a *ranked list* of taxonomic parents with corresponding scores rather than a simple set, based on the weights of the

taxonomic arcs. All other things being equal, a taxonomic parent of an entity (that is not in the same e-component) should receive a greater parent ranking score for that entity if the weight of an incoming arc is higher.

Additionally, to obtain a clean, coherent output, it is crucial to obtain rankings that take into consideration the fact that parents are not independent, but themselves can stand in relationships to each other. For example, two different versions of Wikipedia may have what is essentially the same class (**equals** arcs) or classes that are connected by means of subclass relationships (**subclass** arcs).

This is very important in practice, because we frequently observe that the input arcs link individual articles to their categories, but these categories are language-specific local ones that are not part of a shared multilingual class hierarchy. If an article is found to be in the class **Tal in Trentino-Südtirol** in the German Wikipedia, then the possible parent class **Valley** from WordNet, which is reachable by following **equals** and **subclass** links, should gain further credibility.

The same consideration also applies to the node whose parents are currently being considered. Clearly, when evaluating parents about a Malay Wikipedia article, we may benefit from information available about an equivalent English article entity, and vice versa.

*Property 2 (Dependencies)* A taxonomic arc from a node  $u$  to a node  $v$  with weight greater than 0 should contribute to the ranking scores of nodes  $v'$  that are reachable from  $v$  via **equals** and **subclass** arcs. When evaluating parents for a node  $v_0$ , outgoing taxonomic arcs of nodes  $v'$  that are reachable from  $v_0$  via **equals** arcs should also contribute to the ranking.

Finally, it is fairly obvious that information coming from multiple sources is likely to be more reliable and salient. For example, many Wikipedia editions describe the Colorado River as a river, but only few declare it to be a border of Arizona.

*Property 3 (Aggregation)* If a parent node  $v$  is not in the same e-component as the node  $v_0$  whose parents are being ranked, then, all other things being equal,  $v$  should be given a higher ranking score with incoming taxonomic arcs (of weight greater than 0) from multiple nodes than if  $v$  had incoming arcs from fewer of those nodes.

#### 4.2.2 Markov Chain

Taking these considerations into account, in particular Property 2, requires going beyond conventional rank aggregation algorithms. We use a Markov chain approach that captures dependencies between nodes.

**Definition 6** (Parent Nodes) Given a set of entities  $S$  and a target relation  $r$  (**subclass** or **instance**), the set of *parents*  $P(S, r)$  is the set of all nodes  $v_m$  that are reachable from  $v_0 \in S$  following paths of the form  $(v_0, v_1, \dots, v_m)$  with  $(v_i, v_{i+1}, r_i, w_i) \in A, w_i > 0$  for all  $0 \leq i < m$ , and specific  $r_i$ . The path length  $m$  may be 0 (i.e. the initial entity  $v_0$  is considered part of the parent entity set), and may be limited for practical purposes. When producing subclass arcs as output ( $r = \text{subclass}$ ), all  $r_i$  must be **subclass** or **equals**. When producing instance arcs as output ( $r = \text{instance}$ ), the first  $r_i$  that is not **equals** must be an **instance** relation, and any subsequent  $r_i$  must be either **equals** or **subclass**.



**Definition 7** (Parent e-components) Instead of operating on original sets of parent entities  $P(S, r)$ , we consider the corresponding set of *parent e-components*  $\{E(v) \mid v \in P(S, r)\}$  (see Definition 4), which consists of the e-components for all  $v \in P(S, r)$ .

For every node  $v_0$  in the input graph, we will retrieve the set of possible parents and construct a Markov chain in which each state corresponds to a parent e-component of  $v_0$ . The Markov chain will enable us to create a ranking of those parents.

**Definition 8** (Aggregated Weights) Given a source node  $v_0$  in a knowledge base  $G = (V, A, \Sigma)$ , a target relation  $r$ , and a corresponding set of parent e-components  $\{E_0, \dots, E_n\}$  (such that  $v_0 \in E_0$ ), we define

$$w_{i,j} = \sum_{u \in E_i} \sum_{v \in E_j} \sum_{(u,v,r',w) \in A} w$$

for all  $i, j$  from 0 to  $n$ , where  $r'$  is **instance** if  $i = 0$  and  $r = \text{instance}$ , and  $r'$  is **subclass** in all other cases (i.e. if  $i > 0$  or  $r = \text{subclass}$ ). We further define  $\Gamma_o(i)$  as  $\{j \mid w_{i,j} > 0\}$ .

If the target relation is **subclass**, this definition considers all **subclass** arcs between parent e-components. If the target relation is **instance**, we need to distinguish between outgoing arcs from  $E_0$ , which must be **instance** ones, and other outgoing arcs, which must be **subclass** ones.

**Definition 9** (Markov Chain) Given an entity  $v_0$ , a corresponding set of parent e-components  $\{E_0, \dots, E_n\}$  ( $v_0 \in E_0$ ), a weight matrix  $w_{i,j}$  characterizing the links between different  $E_i$ , and a weight  $c \in \mathbb{R}^+$ , we define a Markov chain  $(E_{i_0}, E_{i_1}, \dots)$  as follows. The set  $\{E_0, \dots, E_n\}$  serves as a finite state space  $S$ , an initial state  $E_{i_0} \in S$  is chosen arbitrarily, and the transition matrix  $Q$  is defined as follows.

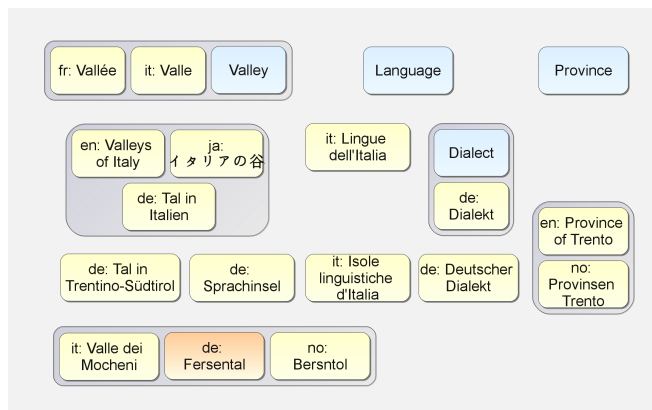
$$Q_{i,j} = \begin{cases} \frac{w_{i,j}}{c + \sum_{k \in \Gamma_o(i)} w_{i,k}} & j \neq 0 \\ \frac{c + w_{i,j}}{c + \sum_{k \in \Gamma_o(i)} w_{i,k}} & j = 0 \end{cases} \quad (3)$$

Figure 5 illustrates a Markov chain defined in this way: Part (a) shows parent e-components corresponding to states, (b) shows state transitions derived from taxonomic arcs between nodes in e-components, and (c) shows how one can transition back to the source node  $E_0$ , which contains **Fersental**, from any state.

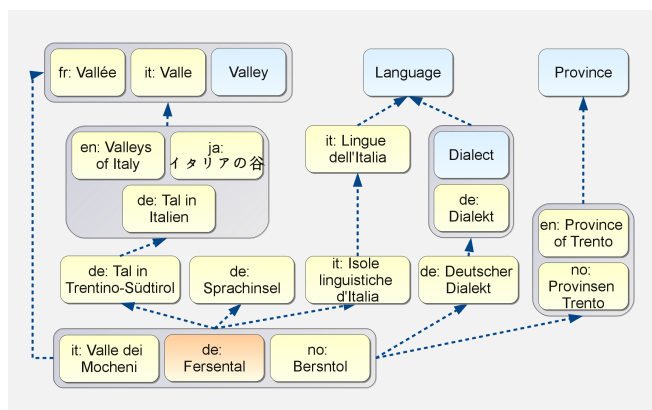
**Theorem 1** A transition matrix  $Q$  as defined in Definition 9 is stochastic.

*Proof* Given  $c > 0$ , for any  $i \in \{0, \dots, n\}$ , we obtain

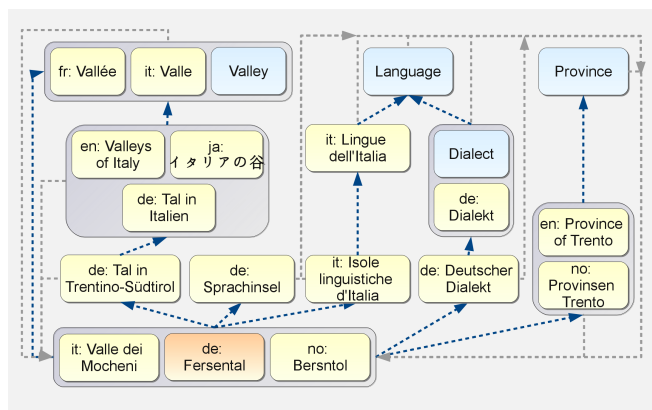
$$\begin{aligned} \sum_{j=0}^n Q_{i,j} &= \frac{c + w_{i,0}}{c + \sum_{k \in \Gamma_o(i)} w_{i,k}} + \sum_{j=1}^n Q_{i,j} \\ &= \frac{c + \sum_{j=0}^n w_{i,j}}{c + \sum_{k=0}^n w_{i,k}} \\ &= 1. \end{aligned}$$



(a) Parent e-components as state space



(b) State transitions based on taxonomic links



(c) Additional state transitions to source node

Fig. 5: Markov chain setup

The state space includes the e-component  $E_0$  containing the source node. The probability mass received by  $E_0$  rather than by genuine parents  $E_i$  with  $i > 0$  in the stationary distribution reflects the extent of our uncertainty about the parents. For instance, if all immediate parents of the source node are linked with very low weights, then  $E_0$  will attract a high probability mass. In the definition,  $c$  is the weight endowed to random restarts, i.e. transitions from arbitrary states back to  $E_0$ . Larger choices of  $c$  lead to a bias towards more immediate parents of  $E_0$ , while lower values work in favour of more general (and presumably more reliable) parents at a higher level. It is easy to see that the Markov chain is irreducible and aperiodic if  $c > 0$ , so a unique stationary distribution must exist in those cases.

**Theorem 2** (*Stationary Probability*) *The Markov chain possesses a unique stationary probability distribution  $\pi$  with  $\pi = \pi Q$ .*

*Proof* For any state  $E \in S$ , there exists some node  $v_m \in E$  that is reachable from the source node  $v_0$  by following a path of statements with non-zero weights as specified in Definition 6. The corresponding weights  $w_{i,j}$  and state transition probabilities  $Q_{i,j}$  along the path must be non-zero. Hence, every state is reachable from  $E_0$ .

Since  $c > 0$ , we obtain a non-zero random restart probability  $Q_{i,0} > 0$  for every  $i$ , so from every state one can transition back to  $E_0$ , and thus the chain is irreducible. Additionally, since  $c > 0$ , the state  $E_0$  is aperiodic (one can remain in  $E_0$  for any amount of steps), and hence the entire chain is aperiodic. By the Fundamental Theorem of Markov chains, a unique stationary distribution exists.

#### 4.2.3 Markov Chain Taxonomy Induction

This implies that we can use the stationary distribution of the Markov chain to rank parents of a source node with respect to their connectedness to that source node. The stationary distribution can easily be computed with the power iteration method. Algorithm 4.1 captures the steps taken to induce the taxonomy.

*Input.* As input, it takes a graph  $G_0$  as defined in Section 2.1, containing information from the original knowledge sources as well as noisy **equals** and taxonomic statements, as produced by Algorithm 3.1. Additionally, one supplies the  $c$  parameter from Definition 9, an output selection function  $\sigma$  discussed below, parameters  $\epsilon, i_{\max}$  for the stationary probability computation, and the taxonomic root node  $v_R$  which is supposed to subsume all other classes (e.g. **Entity**).

*Forming e-components.* The algorithm begins by forming consistent e-components from the output of the equivalence consistency algorithm described in Section 4.1. These become the entities of the output knowledge base. In practice, one may want to create entity identifier strings based on the entity identifiers within the e-component, perhaps preferring article titles in a specific language. Non-taxonomic statements like **means** statements that provide human-readable terms or statements capturing factual knowledge like birth dates of people are directly mapped to the e-components.

**Algorithm 4.1** Markov Chain Taxonomy Induction algorithm

---

```

1: procedure TAXONOMY( $G_0 = (V_0, A_0, \Sigma), c, \sigma, \epsilon, i_{\max}, v_R$ )
2:    $D_0, \dots, D_k \leftarrow$  distinctness assertions for  $G_0$  ▷ cf. Section 4.1
3:   enforce consistency of  $G_0, D_0, \dots, D_k$  ▷ cf. Section 4.1
4:    $V \leftarrow \{E(v) \mid v \in V_0\}$  ▷ consistent e-components become nodes
5:    $\Sigma_T \leftarrow \{\text{equals}, \text{instance}, \text{subclass}\}$  ▷ set of taxonomic relations
6:    $A \leftarrow \{(E(u), E(v), r, w) \mid (u, v, r, w) \in A_0, r \notin \Sigma_T\}$  ▷ map non-taxonomic statements
7:    $A_T \leftarrow \emptyset$ 
8:   for all  $E$  in  $V$  do ▷ for all e-components
9:      $r \leftarrow \begin{cases} \text{subclass} & \text{if } E \text{ likely to be a class} \\ \text{instance} & \text{otherwise} \end{cases}$  ▷ see Section 3.2
10:     $E_0 \leftarrow E$ 
11:     $E_1, \dots, E_n \leftarrow$  enumeration of  $\{E(v) \mid v \in P(E, r)\} \setminus \{E\}$ 
12:    ▷ parent e-components as per Definition 7 in arbitrary order
13:     $Q \leftarrow$  transition matrix for  $E$  using  $E_0, \dots, E_n$  and  $c, r$  ▷ as per Definition 9
14:     $\pi \leftarrow \text{EIGENVECTOR}(Q, \epsilon, i_{\max})$ 
15:     $A_T \leftarrow A_T \cup \{(E, E_i, r, \pi_i) \mid i > 0\}$  ▷ preliminary output
16:    return CLEANING( $V, A_0, A_T, \Sigma \cup \Sigma_T, v_R$ ) ▷ final cleaning (Algorithm 4.2)
17: procedure EIGENVECTOR( $[Q_{i,j}]_{i,j=1,\dots,n}, \epsilon, i_{\max}$ )
18:   choose uniform  $\pi$  with  $\pi_i = \frac{1}{n}$  ▷ initial distribution
19:    $i \leftarrow 0$ 
20:   repeat ▷ Power iteration method
21:      $\pi' \leftarrow \pi$ 
22:      $\pi \leftarrow Q\pi$ 
23:      $i \leftarrow i + 1$ 
24:   until  $\|\pi - \pi'\|_1 < \epsilon$  or  $i \geq i_{\max}$ 
25:   return  $\pi$ 

```

---

*Ranking.* Then, for each e-component  $E$ , the heuristics described in Section 3.2 are used to assess whether  $E$  is likely to be a class (checking headwords for Wikipedia and assuming yes for WordNet synsets without outgoing **instance** arcs). In accordance with the outcome of this assessment, the parents are retrieved and the transition matrix  $Q$  for the Markov chain is constructed. The fixed point  $\pi = \pi Q$  can be computed using a number of different algorithms, e.g. the well-known power iteration method. Although this process needs to be repeated for all e-components, these steps are nevertheless not a bottleneck (see Section 5).

The output knowledge base is generated from this ranking using an additional cleaning algorithm, described below in Section 4.2.5.

#### 4.2.4 Analysis.

Given a knowledge graph  $G = (V_0, A_0, \Sigma)$  stored in a data structure that allows lookups in both directions of a directed arc, e-components can be found in linear time, i.e.  $O(|V_0| + |A_0|)$ , by iterating over the nodes and starting a depth-first search whenever an unseen node is encountered. Due to the overall sparsity of the graph with respect to **equals** arcs, the runtime will tend to be close to  $O(|V_0|)$ . Subsequently, for each  $E \in V$ , the same strategy can be used to retrieve the set of parent e-components, and the weights  $w_{i,j}$  can be computed on the fly while doing this. Computing the Markov chain's transition matrix  $Q$  can take  $O(|V|^2)$  steps, and approximating the stationary distribution requires  $O(|V|^2)$  operations if the power iteration method is used with a constant  $i_{\max}$ . This means that with these implementation choices, the overall worst-case complexity of the algorithm is

$O(|V_0|^3)$ . In practice, the set of parent e-components will be small, and additionally the transition matrices will be sparse, so the algorithm runs fairly quickly, as we show in Section 6.

**Theorem 3** *The Markov Chain Taxonomy Induction algorithm possesses properties 1, 2, and 3, if  $c > 0$ .*

*Proof* Definition 8 implies that, all other things being equal, a higher weight for a taxonomic arc from some  $u \in E_i$  to a parent  $v \in E_j$  will lead to a higher weight  $w_{i,j}$ . We know that  $c > 0$  and additionally assume  $v \notin E_0$  (i.e.  $j \neq 0$ ). Then, by Definition 9,  $Q_{i,j}$  will increase (and at least  $Q_{i,0}$  will decrease). Additionally, from the proof of Theorem 2, we know that  $Q$  is aperiodic and irreducible and hence regular. Due to the monotonicity of the stationary distribution of regular Markov chains [20], the e-component including  $v$  will have a greater probability mass in the new distribution, and Property 1 is fulfilled.

Similarly, given a node  $v'$  reachable from another node  $v$  via **equals** and **subclass** arcs, the state  $E(v')$  must be reachable from  $E(v)$  with non-zero probability, so any taxonomic arc from a node  $u$  to  $v$  also contributes to the ranking of  $v'$ . When evaluating parents for  $v_0$ , nodes  $v'$  that are reachable from  $v_0$  via **equals** arcs are also in  $E_0 = E(v_0)$ , so outgoing taxonomic arcs of  $v'$  contribute to the ranking, and Property 2 is fulfilled.

Finally, Definition 8 implies that, all other things being equal, a parent  $v \in E_j$  with input arcs from multiple children will have a higher sum of incoming weights  $\sum_i w_{i,j}$  than the same parent if it had fewer of those incoming arcs. With  $c > 0$  and assuming  $j \neq 0$ , this also implies a higher  $\sum_i Q_{i,j}$ . The monotonicity of the stationary distribution [20] then implies that Property 3 is satisfied.

With these properties, Markov Chain Taxonomy Induction allows us to aggregate link information from heterogeneous sources, e.g. from multiple editions of Wikipedia, including category and infobox information, and from WordNet. The output is a much more coherent taxonomic knowledge base, similar to the example excerpt in Figure 3, where clean e-components have been merged, and taxonomic links have been aggregated and cleaned. Still, additional cleaning can be performed to obtain an even more consistent taxonomy.

#### 4.2.5 Taxonomy Cleaning

The final clean output taxonomy is generated using Algorithm 4.2 as follows.

- First of all, a selection function  $\sigma$  filters the preliminary output  $A_T$  with respect to application-specific criteria. Usually, this involves enforcing some minimal weight threshold and selecting the top  $k$  parent e-components  $E'$  for a given entity  $E$ . A choice of  $k = 1$  produces a more traditional taxonomy, while higher  $k$  lead to more comprehensive knowledge bases. Other filtering criteria are possible as well, e.g. retaining only parents with Chinese labels or keeping only WordNet synsets as parents.
- In a similar vein, entities that are not connected to the taxonomic root node  $E(v_R)$  (e.g.  $v_R = \text{Entity}$ ) by paths of taxonomic links can be pruned away (the PRUNE procedure), together with their correspondings statements. This leads to an even more coherent knowledge base. Alternatively, these entities could also be made direct children of  $E(v_R)$ .

- The next step involves removing cycles of **subclass** relationships. A cycle of formal subsumptions implies that all entities in the cycle are equivalent. Since we have already merged nodes assumed to be equivalent into e-components, it makes sense to break up such cycles. Cycles can be found in  $O(|V| + |A|)$  steps by determining strongly connected components [80].  $\text{SCC}(V, A)$  is assumed to provide the set of (non-singleton) strongly connected components  $C \in \text{SCC}(V, A)$  with respect to **subclass** links, where each  $C$  is represented as a non-empty set of arcs. The algorithm repeatedly removes the lowest-weighted **subclass** arcs, until no strongly connected components and hence no cycles remain.
- Finally, whenever there is an arc to a parent that is also a higher-order parent, we remove the redundant direct arc to the parent. Formally, this corresponds to computing the smallest graph  $\text{TR}(V, A_T)$  that still has the same closure as  $(V, A_T)$  with respect to **subclass** and **instance**. In the worst case, such transitive reductions may require  $O(|V| |A_T|)$  steps [3], but in practice only a small subset of all nodes serve as parents. This concludes the construction of the final output taxonomy.

---

**Algorithm 4.2** Taxonomy cleaning algorithm

---

```

1: procedure CLEANING( $V, A_0, A_T, \Sigma$ )
2:    $A_T \leftarrow \sigma(A_T)$  ▷ filtering by weight, top- $k$  rank, etc.
3:    $V, A_0, A_T \leftarrow \text{PRUNE}(E(v_R), V, A_0, A_T)$  ▷ remove unconnected branches
4:    $\mathcal{S} \leftarrow \text{SCC}(V, A_T)$  ▷ strongly connected components with respect to subclass
5:   while  $\mathcal{S} \neq \emptyset$  do ▷ remove cycles
6:     choose  $C$  from  $\mathcal{S}$ 
7:      $a \leftarrow \underset{a \in C}{\text{argmin}} w(a)$  ▷ select lowest-weighted subclass arc
8:      $A_T \leftarrow A_T \setminus \{a\}$  ▷ remove  $a$ 
9:      $\mathcal{S} \leftarrow (\mathcal{S} \setminus \{C\}) \cup \text{SCC}(V, C \setminus \{a\})$ 
10:   $A_T \leftarrow \text{TR}(V, A_T)$  ▷ transitive reduction with respect to instance, subclass
11:  return  $G = (V, A_0 \cup A_T, \Sigma)$  ▷ taxonomic knowledge base as output

```

---

## 5 System Architecture

*System.* In order to build MENTA, we developed a platform-independent knowledge base processing framework. For efficiency reasons, the weighted labelled multi-digraphs were stored in custom binary format databases, where we could encode arc labels and weights very compactly. Each entity has an entry in a heavily cached index, with an expected  $O(1)$  look-up time. The entry contains pointers into a large disk-based data file that stores a binary-encoded list of outgoing edges. The number of pointers can be as high as the number of outgoing edges in the worst case, but is only 1 if the data store has been defragmented. The linking heuristics are implemented as functions that assess links between two entities and produce new weights for potential arcs.

*Algorithm Implementation.* The Markov Chain Taxonomy Induction algorithm is used to process the original noisy **subclass** and **instance** arcs that are provided as

input. In order to increase the speed, we limited the maximal parent path length in Definition 6 to  $m = 4$ . This means that thousands of states that would obtain near-zero probabilities are pruned in advance. A second key to making the algorithm run quickly is relying on the fact that many entities share common parents, so the expensive lookups to determine potential parents should be cached. This allowed us to process all 19.9 million e-components (see Section 6) in less than 3 hours on a single 3GHz CPU.

*Scalability* Additionally, since the main loop in Algorithm 4.1 considers each source e-component separately, parallelizing the processing is trivial. The source e-components can be partitioned across multiple machines as well as across multiple processes on each machine. No additional communication is required between them for the Markov Chain ranking.

The pre-processing to create consistent e-components can also be parallelized to a large extent, because each connected component can be processed separately, and whenever a connected component is split into at least two parts, the individual parts can again be processed on separate machines. Additionally, for each individual part, one can also make use of the parallel processing capabilities of recent versions of CPLEX.

*User Interface for Lexical Database Queries.* A simple Web-based user interface has been implemented that allows users to look up words or names and browse some of the multilingual lexical information available in the MENTA knowledge base. Figure 6 provides a screenshot. It is clear that the way language users search for information about words and their meanings has evolved significantly in recent years. Users are increasingly turning to electronic resources to address their lexical information needs because traditional print dictionaries and thesauri take more time to consult and are less flexible with respect to their organization. Alphabetical ordering, for instance, is not well-suited for conveying conceptual and taxonomic relationships between words.

A lexical database like MENTA, in contrast, can simultaneously capture multiple forms of organization and multiple facets of lexical knowledge. In our browsing interface, for a given entity, a list of relevant information is provided, sorted by category, salience and confidence. Especially with the advent of the World Wide Web, users are increasingly expecting to be able to lookup words and choose between different types of information, perhaps navigating quickly from one concept to another based on given links of interest. For example, a non-native speaker of English looking up the word ‘*tercel*’ might find it helpful to see pictures available for the related terms ‘*hawk*’ or ‘*falcon*’. The user can look up a German word like ‘*Tarifautonomie*’, and, despite the lack of a corresponding English Wikipedia article, use MENTA’s taxonomy to find out that it is a sort of judicial principle.

While there have been multilingual interfaces to WordNet-style lexical knowledge in the past [64,5], these provide less than 10 languages as of 2012. The user interface prototype developed for MENTA provides lexical information for more than 200 languages and is available online at <http://www.lexvo.org/uwn/>.




<b>Japanese</b>		
has gloss	jpn: 教員 (きょういん) とは、学校をはじめとする教育施設で、在籍者に対して教育・保育をつかさどる職、または、その職にある者のことである。	
lexicalization	jpn: 教員	
lexicalization	jpn: 先生	
lexicalization	jpn: こうし	
lexicalization	jpn: 教師	
<b>Georgian</b>		
has gloss	kat: განათლების სისტემაში მასწავლებელი არის პიროვნება რომელიც სწავლის პროცესში დახმარებას უწევს, რჩევებს აძლევს მოსწავლეს, ხშირად სამუალო სკოლებში. უმაღლესი სასწავლებლის მასწავლებლებიზოთვის იხმარება სხვა სიტყვები მე. პროფესორი, ლექტორი.	
lexicalization	kat: მასწავლებელი	
lexicalization	kat: პროფესორი	
<b>Central Khmer</b>		
lexicalization	khm: គ្រូ	
<b>Korean</b>		
has gloss	kor: 교사(敎師)는 학생의 배움의 과정에서 이끌어주거나 도움을 주는 사람을 의미. 이러한 행위를 교육이라 부르며, 대부분의 교육은 학교에서 이루어진다. 교사는 스승 또는 선생(先生)이라고도 하며, 대학에서는 교수(敎授)라 부른다. 학생의 반대 의미로서 가르치는 사람들을 통틀어 교수자(敎授者)라고 한다.	
lexicalization	kor: 교사	
lexicalization	kor: 선생	
<b>Kurdish</b>		
lexicalization	kur: مەسۆلم	
<b>Lao</b>		
lexicalization	lao: ຄູ່	
<b>Latin</b>		

Fig. 6: User interface

## 6 Results

We now describe the experimental setup that led to the MENTA knowledge base, as well as several extensions and case studies.

### 6.1 Input Dataset

We wrote a custom Web crawler that downloaded the latest Wikipedia XML dumps from Wikimedia’s download site, retrieving 271 different editions of Wikipedia as of April 2010. The size of the uncompressed XML dumps amounts to around 89.55 GB in total, out of which 25.4 GB stem from the English edition.

### 6.2 Output Analysis

We first analyse the results of the entity equality computation and then study the quality of the taxonomic links, both before and after applying the Markov Chain Taxonomy Induction, in order to demonstrate the contribution of the algorithm. The resulting taxonomy is validated in terms of coherence, accuracy, and coverage. The multilingual lexical knowledge in it is investigated as well.

#### 6.2.1 Entity Equality

*Equality Information.* The linking functions provided 184.3 million directed inter-wiki links and 7.1 million other directed **equals** arcs. The WordNet disambiguation model was obtained by training on 200 out of 407 manually labelled examples, selected randomly among all Wikipedia articles and WordNet synsets sharing a term (increasing the training set size further does not significantly improve the results



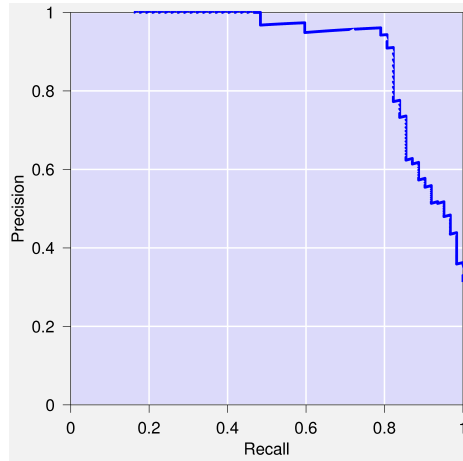


Fig. 7: Precision-recall curve for Wikipedia-WordNet links

because of genuinely hard to disambiguate cases). The precision-recall curve on the remaining 207 examples used as the test set (Fig. 7) shows the remarkably reliable results of the model. With a threshold of 0.5 we obtain 94.3% precision at 80.7% recall ( $F_1$ : 87.0%). The precision only drops sharply once we move towards recall levels significantly above 80%. The overall area under the ROC curve (ROC AUC) is 93.06%.

*Distinctness Information.* The equality arcs led to 19.5 million initial e-components, including templates, categories, and redirects. It turns out that roughly 150,000 of these e-components contained nodes to be separated, among them a single large e-component consisting of nearly 1.9 million nodes. Overall, more than 5.0 million individual node pairs are asserted to be distinct by the distinctness assertions.

*Reconciliation.* We applied the equivalence consistency framework from Section 4.1 to separate the entities and obtain more consistent links. As we did not implement any parallelization in our initial framework, the process took several days to complete, with the expensive linear program solving by CPLEX (for the approximation algorithm) being the major bottleneck. We experimented with agglomerative clustering as an alternative, but found the solutions to be orders of magnitude less optimal in terms of the weights of deleted edges. Using the approximation algorithm, a total of 2.3 million undirected **equals** connections (4.6 million directed arcs) were removed, resulting in 19.9 million e-components after separation.

### 6.2.2 Taxonomy

*Linking Functions.* As additional input to the taxonomy induction algorithm, the linking functions produced what correspond to 1.2 million **subclass** arcs and 20.1 million **instance** arcs between e-components. For the **instance** arcs, we chose

$w_{\text{infobox}} = 2$  because classes derived from infoboxes are more reliable than categories. The WordNet disambiguation model for **subclass** was obtained by training on 1,539 random mappings, the majority of these (1,353) being negative examples. On a test set of 234 random mappings, we obtain a precision of 81.3% at 40.0% recall, however going above 40% recall, the precision drops sharply, e.g. 60.8% precision at 47.7% recall. This task is apparently more difficult than the **equals** disambiguation, because less contextual information is directly available in the category page markup and because our heuristics for detecting classes may fail. Overall, there would be 6.1 million **subclass** arcs, but we applied a minimal threshold weight of 0.4 to filter out the very unreliable ones. The ROC AUC is only 65.8%. This shows that using the original linking functions alone can lead to a taxonomy with many incorrect links.

Table 1: Ranked subclass examples

Class		WordNet Parent	Wikipedia Parent
Science museums in New Mexico	1.	museum	Museums
	2.	science museum	Science museum
	3.	depository	Museums in New Mexico
Cathedrals in Belize	1.	church building	Cathedral
	2.	cathedral (large church)	Churches in Belize
	3.	cathedral (diocese church)	Church buildings
Hamsters	1.	rodent	Rodents
	2.	hamster	Pets
	3.	mammal	Domesticated animals

Table 2: Ranked instance examples

Entity		WordNet Parent	Wikipedia Parent
Fersental	1.	valley	Valleys
	2.	natural depression	Valleys of Italy
	3.	geological formation	Valleys of Trentino / Alto Adige
Cagayan National High School	1.	secondary school	Secondary school
	2.	school	School
	3.	educational institution	High schools in the Philippines
The Spanish Tragedy	1.	book	Book
	2.	publication	British plays
	3.	piece of work	Plays

*Algorithm.* We thus relied on our Markov Chain Taxonomy Induction algorithm to choose reliable parents. In our experiments, the algorithm’s  $c$  parameter was

Table 3: Coverage of individual entities by source Wikipedia

	Instances	Instances Linked to WordNet	Non-English Instances Linked to WN
English	3,109,029	3,004,137	N/A
German	911,287	882,425	361,717
French	868,864	833,626	268,693
Polish	626,798	579,702	159,505
Italian	614,524	594,403	161,922
Spanish	568,373	551,741	162,154
Japanese	544,084	519,153	241,534
Dutch	533,582	508,004	128,764
...	...	...	...
Total	13,982,432	13,405,345	2,917,999
E-components	5,790,490	5,379,832	2,375,695

fixed at  $c = \frac{1}{2}$ , based on the intuition that if there is only one parent with weight 0.5, then that parent should be reached with probability  $\frac{1}{2}$  from the current state. Examples of subclass and instance rankings are given in Tables 1 and 2, respectively, showing the highest-ranked parent entities from WordNet and Wikipedia. Note that in the final output, equivalent parents from WordNet and Wikipedia would in most cases form a single e-component. They are listed separately here for information purposes only.

Out of the 19.9 million e-components in the input, a large majority consist of singleton redirects that were not connected to their redirect targets, due to our careful treatment of redirect links in Section 3.1.

*Coherence.* For roughly 5.8 million e-components, we actually had outgoing **instance** links in the input. To quantify the coherence, we determine what fraction of these e-components can be connected to e-components involving WordNet synsets, as WordNet can be considered a shared upper-level core. Table 3 shows that this succeeds for nearly all e-components. The first column lists the number of entities for which we have outgoing **instance** arcs, while the second column is restricted to those for which we could establish **instance** arcs to WordNet (at a reachability probability threshold of 0.01). The small differences in counts between these two columns indicate that most entities for which there is any class information at all can be integrated into the upper-level backbone provided by WordNet. The third column lists the number of e-components that are independent of the English Wikipedia but have successfully been integrated by our algorithm with **instance** links. While some fraction of those may correspond to entities for which cross-lingual interwiki links need to be added to Wikipedia, large numbers are entities of local interest without any matching English Wikipedia article. Additionally, we found that 338,387 e-components were connected as subclasses of WordNet synsets, out of a total of 360,476 e-components with outgoing **subclass** arcs.

*Accuracy.* Table 4 shows a manual assessment of highest-ranked WordNet-based parent classes for over 100 random entities. The human assessor was shown the name and gloss descriptions of the entity from Wikipedia as well as for the

Table 4: Accuracy of **subclass** arcs to WordNet

top- $k$	Sample Size	Initial Arcs	Ranked Arcs
1	104	82.46% $\pm$ 7.08%	83.38% $\pm$ 6.92%
2	196	57.51% $\pm$ 6.85%	83.03% $\pm$ 5.17%
3	264	45.89% $\pm$ 5.97%	79.87% $\pm$ 4.78%

WordNet-based class and asked to judge whether the entity is an instance of the class (in the specific sense given by the gloss description). We rely on Wilson score intervals at  $\alpha = 0.05$  to generalize our findings to the entire dataset. Wilson score intervals characterize the confidence in a particular proportion. They are much more accurate than standard normal approximations of binomial distributions (Wald intervals), especially when the distribution is skewed, and have been recommended in several comparative studies and analyses [53,17]. For  $k = 2, 3$ , the ranked output is significantly more reliable than the  $w_{i,j}$  between e-components resulting from the initial **subclass** arcs. The aggregation effect is even more noticeable for the **instance** arcs to WordNet in Table 5. To connect instances to WordNet, the algorithm needs to combine **instance** arcs with unreliable **subclass** arcs. Yet, the output is significantly more accurate than the input **subclass** arcs, for  $k = 1, 2$ , and 3. This means that the Markov chain succeeds at aggregating evidence across different potential parents to select the most reliable ones.

We additionally asked speakers of 3 other languages to evaluate the top-ranked WordNet synset for at least 100 randomly selected entities covered in the respective language, but without corresponding English articles. We see that non-English entities are also connected to the shared upper-level ontology fairly reliably. The main sources for errors seem to be topic categories that are interpreted as classes and word sense disambiguation errors from the subclass linking function. Fortunately, we observed that additional manually specified exceptions as in YAGO [76] would lead to significant accuracy improvements with very little effort. Certain categories are very frequent and account for the majority of disambiguation errors.

Table 5: Accuracy of **instance** arcs to WordNet

Language	top- $k$	Sample Size	Wilson Score Interval
English	1	116	90.05% $\pm$ 5.20%
English	2	229	86.72% $\pm$ 4.31%
English	3	322	85.91% $\pm$ 3.75%
Chinese	1	176	90.59% $\pm$ 4.18%
German	1	168	90.15% $\pm$ 4.36%
French	1	151	92.30% $\pm$ 4.06%

*Coverage.* The total number of output e-components in MENTA is roughly 5.4 million excluding redirects (Table 3), so with respect to both the number of entities

Table 6: Multilingual Wordnet (upper-level part of MENTA)

Language	<b>means</b> Statements in MENTA	Distinct Terms in MENTA	Distinct Terms in UWN
Overall	845,210	837,627	822,212
French	36,093	35,699	33,423
Spanish	31,225	30,848	32,143
Portuguese	26,672	26,465	23,499
German	25,340	25,072	67,087
Russian	23,058	22,781	26,293
Dutch	22,921	22,687	30,154

and terms, MENTA is significantly larger than existing multilingual and monolingual taxonomies relying only on the English Wikipedia, which as of February 2011 has around 3.6 million articles. For many of these entities, MENTA contains additional supplementary information extracted from Wikipedia, including short glosses in many different languages, geographical coordinates for countries, cities, places, etc., and links to pictures, videos, and audio clips. For example, when looking up ‘*Mozart*’, pictures as well as audio clips are available.

### 6.2.3 Lexical Knowledge

After forming e-components, the upper-level part of MENTA can be considered a multilingual version of WordNet. A total of 42,041 WordNet synsets have been merged with corresponding Wikipedia articles or categories. We found that WordNet is extended with words and description glosses in 254 languages, although the coverage varies significantly between languages. The average number of Wikipedia-derived labels for these WordNet synsets is 20.

In Table 6, the results are compared with UWN [45], a multilingual wordnet derived mainly from translation dictionaries. While MENTA’s coverage is limited to nouns, we see that MENTA covers comparable numbers of distinct terms. The number of **means** statements is lower than for UWN, because each Wikipedia article is only merged with a single synset. The precision of MENTA’s disambiguation is 94.3%, which is significantly higher than the 85-90% of UWN. This is not surprising, because an approach based on translation dictionaries has much less contextual information available for disambiguation, while MENTA can make use of Wikipedia’s rich content and link structure.

Additionally, MENTA’s output is richer, because we add not only words but also have over 650,000 short description glosses in many different languages as well as hundreds of thousands of links to media files and Web sites as additional information for specific WordNet synsets. Gloss descriptions are not only useful for users but are also important for word sense disambiguation [43]. Finally, of course, our resource adds millions of additional instances in multiple languages, as explained earlier.

## 6.3 Extensions

### 6.3.1 Upper-Level Ontology

As mentioned earlier, the most generic part of an ontological taxonomy, i.e. the part at the top of the hierarchy, is known as the upper-level ontology. In MENTA, we have chosen to retain WordNet as an integral upper-level core.

*Wikipedia as Upper Level.* Alternatively, we may also create a more Wikipedia-centric version where WordNet only serves as background knowledge to help us connect different articles and categories and obtain a more coherent taxonomy. To achieve this, it suffices to have the selection function  $\sigma$  in the algorithm choose only e-components including Wikipedia articles or categories. This amounts to pruning all e-components that consist only of WordNet synsets without corresponding Wikipedia articles or categories. What we obtain is a taxonomy in which the root node is based on the English article **Entity** and its equivalents in other languages. At the upper-most level, the resulting taxonomy is shallower than with WordNet, as many different classes like **Organisms**, **Unit**, **Necessity**, are directly linked to **Entity**. At less abstract levels, the knowledge base becomes more complete. Tables 1 and 2 provide examples of top-ranked parent entities from Wikipedia.

*Alternative Upper-Level Ontologies.* In an additional experiment, we studied replacing WordNet’s lexically oriented upper-level ontology with the more axiomatic one provided by SUMO [54]. SUMO’s expressive first-order (and higher-order) logic axioms enable applications to draw conclusions with some kind of common sense, capturing for example that humans cannot act before being born or that every country has a capital. Extending this with more specific knowledge about entities from Wikipedia can give rise to a fruitful symbiosis, because such axioms can then be applied to individual entities.

We added SUMO’s class hierarchy as well as the publically available mappings between WordNet and SUMO [55] as inputs to the instance ranking, and found that SUMO can be extended with 3,036,146 instances if we accept those linked to a SUMO class with a Markov chain stationary probability of at least 0.01. The sampled accuracy of 177 highest-ranked (top-1) arcs was  $87.9\% \pm 4.7\%$ . The inaccurate links often stemmed from mappings between WordNet and SUMO where the SUMO term did not appear to reflect the word sense from WordNet particularly adequately.

### 6.3.2 Large-Scale Domain-Specific Extensions

A salient feature of our approach is that we can easily tap on additional large-scale knowledge sources in order to obtain even larger knowledge bases. For instance, we can rely on the many domain-specific datasets in the Linked Data Web [12], which describe biomedical entities, geographical objects, books and publications, music releases, etc. In order to integrate them we merely need an **equals** linking function for all individual entities and **equals** or **subclass** arcs for a typically very small number of classes. Our entity aggregation from Section 4.1 will then ensure that the links are consistent, and the Markov Chain Taxonomy Induction algorithm

will choose the most appropriate classes, taking into account the weights of the **subclass** arcs.

As a case study, we investigated a simple integration of the LinkedMDB dataset, which describes movie-related entities. The **equals** links for instances were derived from the existing DBpedia links provided with the dataset, which are available for films and actors. Hence we only needed to specify two manual **equals** arcs for these two classes to allow all corresponding entities to be integrated into MENTA. We obtain additional information on 18,531 films and 11,774 actors already in our knowledge base. Additionally, up to 78,636 new films and 48,383 new actors are added. Similar extensions of MENTA are possible for many other domains by relying on existing third-party datasets.

### 6.3.3 Information Extraction-Based Extensions

Another way of using our algorithm to extend knowledge bases is to rely on textual sources. Pattern-based information extraction approaches are based on the idea of searching a large document collection for strings matching specific textual patterns. For example, the pattern ' $\langle X \rangle$  such as  $\langle Y \rangle$ ' has been found to work well for the **IsA** relation: A matching word sequence like ' $\dots cities such as Paris \dots$ ' allows us to induce statements of the form ('*Paris*', '*City*', **instance**, *w*) [38].

Unfortunately, relying on just a single pattern like the one above leads to very few results. For instance, in a 20 million word New York Times article collection, a well-known study by Hearst found only 46 facts [38]. So-called bootstrapping techniques can be used to discover additional patterns automatically based on a set of examples (e.g. [62]). However, this also tends to imply significantly noisier extracted statements.

In such a situation, our algorithmic framework can serve to select more reliable taxonomic parent words. For example, if '*Paris*' has a number of unreliable parents including '*academy*', '*city*', '*club*', '*town*', then the Markov Chain Taxonomy Induction algorithm, given information about possible superordinate parents, may help us to choose '*municipality*', which generalizes '*city*' and '*town*'.

Noise becomes even more of a problem if we wish to incorporate word sense disambiguation in order to account for the fact that a parent like '*server*' could refer to a waiter or to computing server, among other things. Again, our algorithm can help to choose a single most likely parent for a given word.

We carried out an experiment using 200 textual patterns automatically derived [79] from the Google N-Grams dataset [16] by bootstrapping using seeds from ConceptNet [35]. The derived patterns were applied to the n-gram data to extract large numbers of potential taxonomic facts.

This large set was pre-filtered by requiring that any fact be matched by at least two reliable patterns. We used a set of 63 negative example seeds to filter out unreliable patterns: Any pattern matching any of the negative examples was considered unreliable. This gave us 832,840 **instance** statements for 30,231 distinct source words.

We compared two different methods to select suitable sense-disambiguated parents in WordNet. The baseline method first determines the parents with the highest weights for a given source word (after case normalization and stemming). The weights were computed as  $\exp(n-2)$ , where  $n$  is the number of distinct reliable patterns matched. For the highest-weighted parents, the method chooses the first

noun senses as listed in WordNet as the output. This simple first sense heuristic is known to be extremely competitive in word sense disambiguation tasks, because the most frequent sense has a very high probability of being correct.

The alternative method was to rely on our Markov Chain Taxonomy Induction algorithm. For each word, we created an input knowledge base consisting of the source word and all of its immediate weighted parents, as above for the baseline. Each parent  $t$  was then connected not only to its first noun sense in WordNet, but to all noun senses  $s$ , with weight  $\frac{1}{\text{rank}(t,s)}$  where  $\text{rank}(t,s)$  is the corresponding synset rank (1 for the first noun sense, 2 for the second, and so on). WordNet’s hypernym links were added for parents, but of course care was taken not to include any information from WordNet about the source word itself. We ran our algorithm with  $c = 1$  and then chose the top-ranked WordNet sense.

A manual evaluation of random samples of the two outputs (excluding senses chosen by both algorithms simultaneously) gave us the following results:

- First Sense Baseline:  
24.88%  $\pm$  5.53% precision (sample size: 229)
- Markov Chain Taxonomy Induction:  
80.11%  $\pm$  5.12% precision (sample size: 227)

The results clearly show that Markov Chain Taxonomy Induction succeeds in choosing the right senses by aggregating across individual inputs.

## 6.4 Non-Taxonomic Information

The taxonomic relations provide us with a global structure that connects all entities in the knowledge base. Additionally, we can also include other relationships between entities. First of all, Wikipedia’s category systems in different languages can be used to obtain large numbers of **hasCategory** arcs, connecting entities like **College** to topics like **Education**. Such information can be useful for word sense disambiguation [18]. Earlier, we already mentioned that we can extract geographical coordinates and multimedia links from Wikipedia. Additionally, Wikipedia’s infoboxes provide factual relationships between entities, e.g. the founding year and location of universities, the authors of books, and the genres of musicians. Such information can either be extracted from Wikipedia itself or from other databases that are derived from Wikipedia [6, 76].

## 6.5 Case Studies

### 6.5.1 Entity Search

Knowledge bases like MENTA are useful for semantic search applications. For instance, the Bing Web search engine has relied on Freebase to provide explicit lists of entities for queries like ‘*Pablo Picasso artwork*’.

In Table 7, we compare the numbers of instances obtained as results from the English Wikipedia with the numbers of instances provided by MENTA. The Wikipedia column lists the number of articles belonging to a given category in the English Wikipedia, while the MENTA columns list the number of e-components



Table 7: Entity search query examples

Query	Wikipedia	MENTA (English Wikipedia)	MENTA (All)
cities and towns in Italy	8,156	8,509	12,992
europaen newspapers	13	389	1,963
people	441,710	882,456	1,778,078
video games developed in Japan	832	775	1,706

Table 8: Integrated non-English entities

Wikipedia edition	Entity	Top-Ranked Class in WordNet
French	Guillaume II (évêque de Meaux)	bishop
French	Hansalim	social movement
French	Tropanol	chemical compound
Chinese	王恩	person
Chinese	九巴士893	travel route
Chinese	东京梦华录	book

with outgoing **instance** arcs to the respective class e-components in MENTA’s aggregated ranking (with a minimum stationary probability  $\pi_i$  of 0.01). Even if we consider only MENTA instances present in the English Wikipedia, i.e. e-components that include English Wikipedia pages, we often find more instances than directly given in the English Wikipedia, because our approach is able to infer new parents of instances based on evidence in non-English editions. Table 8 provides examples of entities from non-English Wikipedia editions integrated into the taxonomy.

Machine-readable knowledge bases allow for more advanced expert queries than standard text keyword search. For instance, one could search for philosophers who were also physicists, perhaps born in a specific time period and geographical area.

### 6.5.2 Fine-Grained Named Entity Recognition

Named entity recognition is a standard subtask in many natural language processing systems, which aims at identifying mentions of entities in a text [50]. For instance, a named entity recognition system may attempt to identify all people, companies, organizations, and places mentioned in a stream of news articles.

Standard systems only distinguish between very high-level categories. Typically, these are: Persons, Locations, Organizations and Miscellaneous. MENTA enables a much more fine-grained classification of entity names occurring in a text. For instance, one can attempt to recognize all names of actors or names of rivers mentioned in a text document.

To accomplish this, one feeds the text to a so-called Wikification system [39] in order to detect and disambiguate entity mentions. Subsequently, one consults MENTA to determine the relevant taxonomic classes of the encountered entities.

The evaluation in Section 6.2.2 showed that top-ranked WordNet-level classes for the named entities from Wikipedia have an accuracy around 85 to 90%.

### 6.5.3 Cross-Lingual Image Search

Since much of the content on the Web is written in English, retrieving images based on English keywords usually works well. For instance, an English speaker can easily retrieve images for the keyword ‘*Mottled Duck*’, which refers to a specific species of ducks. A Hungarian speaker searching for the equivalent Hungarian term ‘*Floridai réce*’ will find far fewer results.

MENTA can help here by providing images extracted from articles in other language. For instance, as of April 2012, four images can be extracted from the Swedish Wikipedia article for Mottled Ducks.

MENTA organizes all of these images in a taxonomic hierarchy with sense distinctions. For example, for the keyword ‘*tercel*’, Google’s Image Search shows virtually only images of Toyota Tercel cars, but not images of the bird meaning of ‘*tercel*’. MENTA distinguishes ‘*tercel*’ from ‘*Toyota Tercel*’ by having separate entries (entities) for them. Additionally, MENTA’s taxonomy allows providing the user with pictures available for more general terms like ‘*hawk*’ or sibling terms like ‘*falcon*’.

This means that MENTA with its WordNet-based taxonomy can serve as a replacement for the well-known image classification resource ImageNet [23]. MENTA provides images with open-source-compatible licenses, while ImageNet consists of copyrighted images from the Web and hence cannot be freely distributed.

### 6.5.4 Lexical Gaps in Machine Translations

Modern machine translation systems depend a lot on the amount of available data. Traditional rule-based systems need translation lexicons that are large enough to have entries for all the words that need to be translated and statistical machine translation systems need huge parallel corpora covering the relevant words and phrases in the input documents.

For the vast majority of language pairs, however, the amount of available data of this particular form unfortunately remains very limited. Out-of-vocabulary errors or words that are simply left untranslated are a frequent result. Google Translate for instance, cannot properly translate the sentence ‘*Occasionally there is discussion about how the sackbut should be held*’ to languages like Portuguese or Chinese, because ‘*sackbut*’ is a rare word, referring to an early version of the trombone musical instrument.

MENTA not only provides translations of ‘*sackbut*’ to certain other languages (including Portuguese), but also helps solve lexical gap issues by providing links to superordinate terms in the taxonomy. In this case, MENTA reports that a sackbut is a sort of trombone, and provides numerous words for trombones in many different languages, including Chinese. With such knowledge, machine translation systems can provide translations that are much closer to the original meaning than the mistranslations that Google Translate currently provides.

## 7 Related Work

We now provide an overview of related research efforts and explain how our approach differs from this previous work.

*Mining Wikipedia.* A number of projects have imported basic information from Wikipedia, e.g. translations and categories [40, 70], or simple facts like birth dates, e.g. in Freebase [13]. Such resources lack the semantic integration of conflicting information as well as the taxonomic backbone that is the focus of our work.

Apart from such facts, DBpedia [6] also provides an ontology, based on a set of manually specified mappings from Wikipedia’s infobox templates to a coarse-grained set of 260 classes. However, the majority of English articles do not have any such infobox information, and non-English articles without English counterparts are mostly ignored. DBpedia additionally includes class information from YAGO [76], a knowledge base that links entities from Wikipedia to an upper-level ontology provided by WordNet. We adopted this idea of using WordNet as background knowledge as well as some of the heuristics for creating instance and subclass arcs. YAGO’s upper ontology is entirely monolingual, while in MENTA the class hierarchy itself is also multilingual and additionally accommodates entities that are found in non-English Wikipedias. Furthermore, the class information is simultaneously computed from multiple editions of Wikipedia. Nastase et al. [51] exploit categories not only to derive *isA* relationships, but also to uncover other types of relations, e.g. a category like ‘*Universities in Milan*’ also reveals where a university is located.

*Linking Heuristics.* Numerous generic heuristics have been proposed to link equivalent entities – Dorneles et al. provide a survey [25]. In general, any such heuristic can be used to produce equivalence information serving as input to our taxonomy induction algorithm.

Only a few other projects have proposed heuristics specifically optimized for interlinking Wikipedia editions or linking Wikipedia to WordNet. Ponzetto et al. [66, 65] studied heuristics and strategies to link Wikipedia categories to parent categories and to WordNet. Their results are significant, as they lead to a taxonomy of classes based on the category system of the English Wikipedia, however they did not study how to integrate individual entities (articles) into this taxonomy.

Recently, Navigli & Ponzetto [52] investigated matching English Wikipedia articles with WordNet synsets by comparing the respective contextual information, obtaining a precision of 81.9% at 77.5% recall. Wu & Weld [84] use parsing and machine learning to link infobox templates to WordNet. The Named Entity WordNet project [82] attempts to link entities from Wikipedia as instances of roughly 900 WordNet synsets. Others examined heuristics to generate new cross-lingual links between different editions of Wikipedia [57, 74].

The focus in our work is on a suitable algorithmic framework to aggregate and rank information delivered by such heuristics, and many of these heuristics could in fact be used as additional inputs to our algorithm. The same holds for the large body of work on information extraction to find taxonomic *isA* relationships in text corpora [38, 72, 28, 33, 77], machine-readable dictionaries [49], or search engine query logs [7]. Adar et al. [1] and Bouma et al. [15] studied how information from

one Wikipedia’s infoboxes can be propagated to another edition’s articles, which is distinct from the problem we are tackling.

*Multilingual Knowledge Bases.* Concerning multilingual knowledge bases in general, previous results have been many orders of magnitude smaller in terms of the number of entities covered [42,32], or lack an ontological class hierarchy [44]. EuroWordNet [83] provides multilingual labels for many general words like ‘*university*’, but lacks the millions of individual named entities (e.g. ‘*Napa Valley*’ or ‘*San Diego Zoo*’) that Wikipedia provides. The largest comparable resources are BabelNet [52] and WikiNet [51]. These were developed in parallel to MENTA and also draw on Wikipedia, but have extracted other types of information rather than aiming at integrating all Wikipedia editions into a single taxonomy. Fortunately, Wikipedia-based identifiers can serve as a common ground to use all of these resources simultaneously.

*Taxonomy Induction Algorithms.* Hierarchical agglomerative clustering has been used to derive monolingual taxonomies [41], however clustering techniques will often merge concepts based on semantic relatedness rather than specific ontological relationships. Our work instead capitalizes on the fact that reasonably clean upper ontologies already exist, so the main challenge is integrating the information into a coherent whole. There are numerous studies on supervised learning of hierarchical classifications [26], but such approaches would require reliable training data for each of the several hundred thousand classes that we need to consider. Another interesting alternative, proposed by Wu & Weld [84], is to rely on Markov Logic Networks to jointly perform mappings between entities and derive a taxonomy. Unfortunately, such techniques do not scale to the millions of entities we deal with in our setting.

Snow et al. [73] proposed a monolingual taxonomy induction approach that considers the evidence of coordinate terms when disambiguating. Their approach assumes that evidence for any superordinate candidates is directly given as input, while our approach addresses the question of how to produce evidence for superordinate candidates based on evidence for subordinate candidates. For instance, very weak evidence that Stratford-upon-Avon is either a village or perhaps a city may suffice to infer that it is a populated place. Talukdar et al. [78] studied a random walk technique to propagate class labels from seed instances to other coordinate instances, but did not consider hierarchical dependencies between classes.

*Taxonomic Data Integration.* There has been a large amount of research on aligning two taxonomic resources [30]. On et al. [58] studied how to best group together sets of equivalent entities, without however additionally taking into account explicit criteria for distinctness as our approach allows. Unfortunately, finding equivalent items is only one of several steps when aiming at merging and integrating taxonomies. In many cases, an item from one resource does not have any equivalent in the other resource, but instead only a sub- or superordinate item.

A few systems have been developed that address this problem of merging in a semi-automatic way, by requiring human experts to assist in merging the resources [75,56]. Thau et al. [81] described an automatic algebraic framework for taxonomy merging. This approach assumes that all input mappings are 100% correct, and the output is formally equivalent to the union of both input taxonomies. In contrast,

our own approach considers a very different scenario, allowing for links between items to be weighted and in fact to be unreliable. Ponzetto & Navigli [65] proposed a method to restructure a taxonomy based on its agreement with a more reliable taxonomy (WordNet), but do not address how to integrate multiple taxonomies. Raunich & Rahm [67] developed a system that integrates one ontology into another by removing cycles and other redundant or irrelevant links.

None of these approaches aim at aggregating evidence from multiple sources and producing a ranking based on the available evidence. Furthermore, none of the above approaches address the intricate task of merging evidence from more than just two input data sources, especially when there are millions of input links connecting them in various ways.

*Markov Chains.* Our Markov Chain Taxonomy Induction algorithm is most similar to PageRank with personalized random jump vectors [61, 36]; however our transition matrix is based on statement weights, and the probability for jumping to a start node of a random walk depends on the weights of the alternative statements rather than being uniform for all nodes. Uniform weights mean that single parents are visited with very high probability even if they are only very weakly connected, while in our approach such irrelevant parents will not obtain a high transition probability. Other studies have relied on PageRank to find important vocabulary in an ontology [85] and to perform word sense disambiguation [47]. Our Markov chain model differs from these in that we aim at identifying salient parents for a specific node rather than generic random walk reachability probabilities. We are not aware of any Markov chain-based approaches for constructing class hierarchies.

## 8 Conclusion

We have presented techniques to relate entities from multiple knowledge sources to each other in terms of a coherent taxonomic hierarchy. As a first step, this involves using linking functions to connect individual nodes that are equivalent or stand in a taxonomic relationship to each other. Subsequently, we use distinctness heuristics and graph algorithms to clean up the `equals` links. Finally, a Markov chain ranking algorithm is used to produce a much more coherent taxonomy while taking into account arc weights, dependencies in terms of `equals` arcs, and higher-order parents, among other things.

These methods were applied to the task of combining over 200 language-specific editions of Wikipedia as well as WordNet into a single knowledge base, where we succeeded in integrating 13.4 million out of 14.0 million possible articles from different Wikipedia editions into the upper-level ontology. The result of this work is MENTA, presumably the largest taxonomically organized multilingual lexical knowledge base, which is freely available for download at <http://www.mpi-inf.mpg.de/yago-naga/menta/>.

We believe that MENTA can support a number of semantic applications, which leads to several opportunities for new research. For instance, all-words word sense disambiguation using WordNet is well-studied but definitely not a solved problem [2]. In particular, established systems have not been designed to support large

numbers of named entities in conjunction with WordNet’s fine-grained sense distinctions. Additionally, many current systems need to be adapted to operate on non-English text.

The entity search problem also needs to be studied further. Users may wish to pose natural language queries like ‘*What are the top-selling video games developed in Japan?*’ or ‘*Which cities in France have mayors born in the 1930s?*’. The required factual data from Wikipedia can be incorporated into MENTA, but mapping natural language requests to knowledge base queries is non-trivial.

Further experiments could be carried out by applying our taxonomy induction in alternative settings. Apart from MENTA, we showed that our Markov Chain Taxonomy Induction algorithm is flexible enough to work with an alternative upper-level ontology like SUMO, with additional knowledge from the Linked Data Web, or with information extraction systems that collect named entities and clues about their classes from text. Overall, this framework paves the way for new knowledge bases that integrate many existing large-scale data sources while offering more than the sum of the inputs.

## References

1. Adar, E., Skinner, M., Weld, D.S.: Information arbitrage across multi-lingual Wikipedia. In: Proc. WSDM 2009. ACM, New York, NY, USA (2009)
2. Agirre, E., López de Lacalle, O., Fellbaum, C., Hsieh, S.K., Tesconi, M., Monachini, M., Vossen, P., Segers, R.: SemEval-2010 task 17: All-words word sense disambiguation on a specific domain. In: Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 75–80. ACL, Uppsala, Sweden (2010)
3. Aho, A.V., Garey, M.R., Ullman, J.D.: The transitive reduction of a directed graph. SIAM Journal on Computing **1**(2), 131–137 (1972)
4. Atserias, J., Rigau, G., Villarejo, L.: Spanish WordNet 1.6: Porting the Spanish WordNet across Princeton versions. In: Proceedings of the 4th Language Resources and Evaluation Conference (LREC 2004) (2004)
5. Atserias, J., Villarejo, L., Rigau, G., Agirre, E., Carroll, J., Magnini, B., Vossen, P.: The MEANING multilingual central repository. In: Proc. GWC 2004 (2004)
6. Auer, S., Bizer, C., Lehmann, J., Kobilarov, G., Cyganiak, R., Ives, Z.: DBpedia: A nucleus for a web of open data. In: Proc. ISWC/ASWC, LNCS 4825. Springer (2007)
7. Baeza-Yates, R., Tiberi, A.: Extracting semantic relations from query logs. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007), pp. 76–85. ACM, New York, NY, USA (2007). DOI <http://doi.acm.org/10.1145/1281192.1281204>
8. Bast, H., Chitea, A., Suchanek, F., Weber, I.: ESTER: Efficient search in text, entities, and relations. In: Proc. SIGIR. ACM, New York, NY, USA (2007)
9. Bellaachia, A., Amor-Tijani, G.: Enhanced query expansion in English-Arabic CLIR. In: Proc. DEXA 2008. IEEE Computer Society, Washington, DC, USA (2008). DOI <http://dx.doi.org/10.1109/DEXA.2008.52>
10. Benitez, L., Cervell, S., Escudero, G., Lopez, M., Rigau, G., Taulé, M.: Methods and tools for building the Catalan WordNet. In: Proceedings of the ELRA Workshop on Language Resources for European Minority Languages at LREC 1998 (1998)
11. Bishop, C.M.: Pattern Recognition and Machine Learning, 1st ed. corr. 2nd printing edn. Springer (2007)
12. Bizer, C., Heath, T., Berners-Lee, T.: Linked data – the story so far. Int. J. Sem. Web and Inform. Sys. (2009)
13. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: A collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD 2008), pp. 1247–1250. ACM, New York, NY, USA (2008). DOI <http://doi.acm.org/10.1145/1376616.1376746>

14. Bouamrane, M.M., Rector, A., Hurrell, M.: Using owl ontologies for adaptive patient information modelling and preoperative clinical decision support. *Knowledge and Information Systems* **29**, 405–418 (2011). URL <http://dx.doi.org/10.1007/s10115-010-0351-7>
15. Bouma, G., Duarte, S., Islam, Z.: Cross-lingual alignment and completion of Wikipedia templates. In: *Proc. Workshop Cross Lingual Information Access*. ACL (2009)
16. Brants, T., Franz, A.: *Web 1T 5-gram Version 1*. Linguistic Data Consortium, Philadelphia, PA, USA (2006)
17. Brown, L.D., Cai, T.T., DasGupta, A.: Interval estimation for a binomial proportion. *Statistical Science* **16**(2), 101–133 (2001)
18. Buitelaar, P., Magnini, B., Strapparava, C., Vossen, P.: Domains in sense disambiguation. In: E. Agirre, P. Edmonds (eds.) *Word Sense Disambiguation: Algorithms and Applications*, chap. 9, pp. 275–298. Springer (2006)
19. Chen, C.L., Tseng, F., Liang, T.: An integration of fuzzy association rules and wordnet for document clustering. *Knowledge and Information Systems* **28**, 687–708 (2011). URL <http://dx.doi.org/10.1007/s10115-010-0364-2>
20. Chien, S., Dwork, C., Kumar, R., Simon, D.R., Sivakumar, D.: Link evolution: Analysis and algorithms. *Internet Mathematics* **1**(3) (2003)
21. Cook, D.: MLSN: A multi-lingual semantic network. In: *Proc. NLP* (2008)
22. Davis, M., Dürst, M.: Unicode normalization forms, Rev. 29. Tech. rep., Unicode (2008). URL <http://unicode.org/reports/tr15/>
23. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Li, F.F.: ImageNet: A large-scale hierarchical image database. In: *Proc. CVPR 2009*, pp. 248–255. IEEE (2009)
24. Dhillon, I.S., Guan, Y., Kulis, B.: Weighted graph cuts without eigenvectors. a multilevel approach. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(11), 1944–1957 (2007). DOI <http://dx.doi.org/10.1109/TPAMI.2007.1115>
25. Dorneles, C., Gonçalves, R., dos Santos Mello, R.: Approximate data instance matching: a survey. *Knowledge and Information Systems* **27**, 1–21 (2011). URL <http://dx.doi.org/10.1007/s10115-010-0285-0>
26. Dumais, S.T., Chen, H.: Hierarchical classification of Web content. In: *Proc. SIGIR*. ACM, Athens, Greece (2000)
27. Edmonds, J., Karp, R.M.: Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM* **19**(2), 248–264 (1972). DOI <http://doi.acm.org/10.1145/321694.321699>
28. Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Web-scale information extraction in KnowItAll: Preliminary results. In: *Proceedings of the 13th International World Wide Web Conference (WWW 2004)*, pp. 100–110 (2004)
29. Etzioni, O., Reiter, K., Soderland, S., Sammer, M.: Lexical translation with application to image search on the web. In: *Proc. MT Summit XI* (2007)
30. Euzenat, J., Shvaiko, P.: *Ontology matching*. Springer-Verlag, Heidelberg, Germany (2007)
31. Fellbaum, C. (ed.): *WordNet: An Electronic Lexical Database*. The MIT Press (1998)
32. Fellbaum, C., Vossen, P.: Connecting the universal to the specific: Towards the global grid. In: *Proc. IWIC, LNCS*, vol. 4568. Springer (2007)
33. Garera, N., Yarowsky, D.: Minimally supervised multilingual taxonomy and translation lexicon induction. In: *Proc. IJCNLP* (2008)
34. Gong, Z., Cheang, C.W., U, L.H.: Web query expansion by WordNet. In: *Proc. DEXA 2005, LNCS*, vol. 3588. Springer (2005)
35. Havasi, C., Speer, R., Alonso, J.: ConceptNet 3: a flexible, multilingual semantic network for common sense knowledge. In: *Proc. RANLP 2007*. Borovets, Bulgaria (2007)
36. Haveliwala, T.H.: Topic-sensitive PageRank. In: *Proceedings of the 11th International World Wide Web Conference (WWW 2002)* (2002)
37. Hayes, P.: RDF semantics. W3C recommendation, World Wide Web Consortium (2004). URL <http://www.w3.org/TR/2004/REC-rdf-mt-20040210/>
38. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: *Proceedings of the 14th Conference on Computational Linguistics (COLING 1992)*, pp. 539–545. ACL, Morristown, NJ, USA (1992). DOI <http://dx.doi.org/10.3115/992133.992154>
39. Hoffart, J., Yosef, M.A., Bordino, I., Fürstena, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., Weikum, G.: Robust disambiguation of named entities in text. In: *Proc. EMNLP 2011*, pp. 782–792. ACL (2011)

40. Kinzler, D.: Automatischer Aufbau eines multilingualen Thesaurus durch Extraktion semantischer und lexikalischer Relationen aus der Wikipedia. Master's thesis, Universität Leipzig (2008)
41. Klapaftis, I.P., Manandhar, S.: Taxonomy learning using word sense induction. In: Proc. NAACL-HLT. ACL (2010)
42. Knight, K., Luk, S.K.: Building a large-scale knowledge base for machine translation. In: Proc. AAAI (1994)
43. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In: Proc. SIGDOC 1986. ACM (1986). DOI <http://doi.acm.org/10.1145/318723.318728>
44. Mausam, Soderland, S., Etzioni, O., Weld, D., Skinner, M., Bilmes, J.: Compiling a massive, multilingual dictionary via probabilistic inference. In: Proc. ACL-IJCNLP. ACL (2009)
45. de Melo, G., Weikum, G.: Towards a universal wordnet by learning from combined evidence. In: Proc. CIKM 2009. ACM, New York, NY, USA (2009). DOI <http://doi.acm.org/10.1145/1645953.1646020>
46. de Melo, G., Weikum, G.: Untangling the cross-lingual link structure of wikipedia. In: Proc. ACL 2010. ACL, Uppsala, Sweden (2010). URL <http://www.aclweb.org/anthology/P10-1087>
47. Mihalcea, R., Tarau, P., Figa, E.: PageRank on semantic networks, with application to word sense disambiguation. In: Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004), p. 1126. ACL, Morristown, NJ, USA (2004). DOI <http://dx.doi.org/10.3115/1220355.1220517>
48. Milne, D.N., Witten, I.H., Nichols, D.M.: A knowledge-based search engine powered by Wikipedia. In: Proc. CIKM 2007. ACM, New York, NY, USA (2007). DOI <http://doi.acm.org/10.1145/1321440.1321504>
49. Montemagni, S., Vanderwende, L.: Structural patterns vs. string patterns for extracting semantic information from dictionaries. In: Proceedings of the 14th Conference on Computational Linguistics (COLING 1992), pp. 546–552. ACL, Morristown, NJ, USA (1992). DOI <http://dx.doi.org/10.3115/992133.992155>
50. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. *Linguisticae Investigationes* **30**(1), 3–26 (2007). DOI 10.1075/li.30.1.03nad. URL <http://dx.doi.org/10.1075/li.30.1.03nad>
51. Nastase, V., Strube, M.: Transforming wikipedia into a large scale multilingual concept network. *Artificial Intelligence* (0), – (2012). DOI 10.1016/j.artint.2012.06.008. URL <http://www.sciencedirect.com/science/article/pii/S0004370212000781>
52. Navigli, R., Ponzetto, S.P.: Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence* **193**(0), 217 – 250 (2012). DOI 10.1016/j.artint.2012.07.001. URL <http://www.sciencedirect.com/science/article/pii/S0004370212000793>
53. Newcombe, R.G.: Two-sided confidence intervals for the single proportion: comparison of seven methods. *Statistics in Medicine* **17**(8), 857–872 (1998)
54. Niles, I., Pease, A.: Towards a Standard Upper Ontology. In: Proc. FOIS (2001)
55. Niles, I., Pease, A.: Linking lexicons and ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. In: Proceedings of the IEEE International Conference on Information and Knowledge Engineering (IKE 2003), pp. 412–416 (2003)
56. Noy, N.F., Musen, M.A.: The prompt suite: interactive tools for ontology merging and mapping. *Int. J. Hum.-Comput. Stud.* **59**(6) (2003)
57. Oh, J.H., Kawahara, D., Uchimoto, K., Kazama, J., Torisawa, K.: Enriching multilingual language resources by discovering missing cross-language links in Wikipedia. In: Proc. WI/IAT. IEEE, Washington, DC, USA (2008). DOI <http://dx.doi.org/10.1109/WIAT.2008.317>
58. On, B.W., Lee, I., Lee, D.: Scalable clustering methods for the name disambiguation problem. *Knowledge and Information Systems* **31**, 129–151 (2012). URL <http://dx.doi.org/10.1007/s10115-011-0397-1>. DOI 10.1007/s10115-011-0397-1
59. Orav, H., Vider, K.: Estonian Wordnet and lexicography. In: Proceedings of the 11th International Symposium on Lexicography. Copenhagen (2005)
60. Ordan, N., Wintner, S.: Hebrew WordNet: a test case of aligning lexical databases across languages. *International Journal of Translation* **19**(1) (2007)
61. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab (1999). URL <http://ilpubs.stanford.edu:8090/422/>

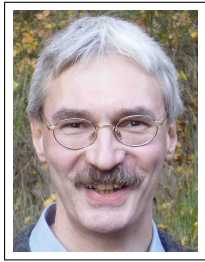


62. Pantel, P., Pennacchiotti, M.: Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In: Proc. ACL 2006. ACL (2006)
63. Pasternack, J., Roth, D.: Learning better transliterations. In: Proc. CIKM 2009. ACM, New York, NY, USA (2009). DOI <http://doi.acm.org/10.1145/1645953.1645978>
64. Pianta, E., Bentivogli, L., Girardi, C.: MultiWordNet: Developing an aligned multilingual database. In: Proc. GWC (2002)
65. Ponzetto, S.P., Navigli, R.: Large-scale taxonomy mapping for restructuring and integrating Wikipedia. In: Proc. IJCAI. Morgan Kaufmann (2009)
66. Ponzetto, S.P., Strube, M.: WikiTaxonomy: A large scale knowledge resource. In: Proc. ECAI 2008. IOS Press (2008)
67. Raunich, S., Rahm, E.: Atom: Automatic target-driven ontology merging. In: Proceedings of the 2011 IEEE 27th International Conference on Data Engineering, ICDE '11. IEEE Computer Society, Washington, DC, USA (2011). DOI 10.1109/ICDE.2011.5767871. URL <http://dx.doi.org/10.1109/ICDE.2011.5767871>
68. Rodríguez, H., Farwell, D., Farreres, J., Bertran, M., Alkhalifa, M., Martí, M.A., Black, W.J., Elkateb, S., Kirk, J., Pease, A., Vossen, P., Fellbaum, C.: Arabic WordNet: Current state and future extensions. In: Proceedings of the 4th Global WordNet Conference (GWC 2008) (2008)
69. Schlaefel, N., Ko, J., Betteridge, J., Pathak, M., Nyberg, E., Sautter, G.: Semantic extensions of the Ephyra QA system for TREC 2007. In: Proc. TREC 2007. NIST (2007)
70. Silberger, C., Wentland, W., et al.: Building a multilingual lexical resource for named entity disambiguation, translation and transliteration. In: Proc. LREC. ELRA (2008)
71. Sleator, D., Temperley, D.: Parsing English with a Link Grammar. In: Proceedings of the 3rd International Workshop on Parsing Technologies (1993)
72. Snow, R., Jurafsky, D., Ng, A.Y.: Learning syntactic patterns for automatic hypernym discovery. In: Advances in Neural Information Processing Systems 17 (NIPS 2004) (2004)
73. Snow, R., Jurafsky, D., Ng, A.Y.: Semantic taxonomy induction from heterogenous evidence. In: Proc. ACL. ACL, Morristown, NJ, USA (2006). DOI <http://dx.doi.org/10.3115/1220175.1220276>
74. Sorg, P., Cimiano, P.: Enriching the crosslingual link structure of Wikipedia. A classification-based approach. In: Proc. AAAI 2008 Workshop Wikipedia and AI (2008)
75. Stumme, G., Maedche, A.: FCA-MERGE: bottom-up merging of ontologies. In: Proc. 17th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI 2001. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001)
76. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A core of semantic knowledge. In: Proc. WWW. ACM (2007)
77. Sánchez, D., Isern, D., Millan, M.: Content annotation for the semantic web: an automatic web-based approach. Knowledge and Information Systems **27**, 393–418 (2011). URL <http://dx.doi.org/10.1007/s10115-010-0302-3>. 10.1007/s10115-010-0302-3
78. Talukdar, P.P., Reisinger, J., Paşca, M., Ravichandran, D., Bhagat, R., Pereira, F.: Weakly-supervised acquisition of labeled class instances using graph random walks. In: Proc. EMNLP 2008. ACL, Morristown, NJ, USA (2008)
79. Tandon, N., de Melo, G.: Information extraction from web-scale n-gram data. In: C. Zhai, D. Yarowsky, E. Viegas, K. Wang, S. Vogel (eds.) Proc. Web N-gram Workshop at ACM SIGIR 2010, vol. 5803. ACM (2010)
80. Tarjan, R.: Depth-first search and linear graph algorithms. SIAM Journal on Computing **1**(2), 146–160 (1972)
81. Thau, D., Bowers, S., Ludäscher, B.: Merging taxonomies under rcc-5 algebraic articulations. In: Proc. 2nd International Workshop on Ontologies and Information Systems for the Semantic Web (ONISW 2008), pp. 47–54. ACM, New York, NY, USA (2008). DOI 10.1145/1458484.1458492. URL <http://doi.acm.org/10.1145/1458484.1458492>
82. Toral, A., Muñoz, R., Monachini, M.: Named Entity WordNet. In: Proc. LREC. ELRA (2008)
83. Vossen, P. (ed.): EuroWordNet: A Multilingual Database with Lexical Semantic Networks. Springer (1998)
84. Wu, F., Weld, D.S.: Automatically refining the Wikipedia infobox ontology. In: Proc. WWW. ACM (2008). DOI <http://doi.acm.org/10.1145/1367497.1367583>
85. Zhang, X., Li, H., Qu, Y.: Finding important vocabulary within ontology. In: Proc. ASWC 2006, LNCS, vol. 4185. Springer (2006)

## Author Biographies



**Gerard de Melo** is a Visiting Scholar at UC Berkeley working in the Artificial Intelligence group of the International Computer Science Institute (ICSI). Prior to that, he was a member of the Max Planck Institute for Informatics and received his doctoral degree from Saarland University with distinction in 2010. He has published several award-winning papers on interdisciplinary research spanning topics like web mining, natural language processing, data integration, and graph algorithms. He also maintains the *lexvo.org* site, which is used by many providers of Linked Data.



**Gerhard Weikum** is a Scientific Director at the Max Planck Institute for Informatics in Saarbrücken, Germany, where he is leading the Databases and Information Systems department. Earlier he held positions at Saarland University, ETH Zurich, and MCC in Austin, Texas, and was a visiting senior researcher at Microsoft Research Redmond. He co-authored a comprehensive textbook on transactional systems, and has worked on distributed systems, self-tuning database systems, DB&IR integration, and automatic knowledge harvesting from Web and text sources. Gerhard Weikum is an ACM Fellow and received the VLDB 10-Year Award. From 2003 through 2009 he was president of the VLDB Endowment.