# Leveraging the Crowdsourcing of Lexical Resources for Bootstrapping a Linguistic Data Cloud

Sebastian Hellmann*, Jonas Brekle*, and Sören Auer

Universität Leipzig, Institut für Informatik, AKSW,
Postfach 100920, D-04009 Leipzig, Germany
{hellmann,brekle,auer}@informatik.uni-leipzig.de
http://aksw.org

**Abstract.** We present a declarative approach implemented in a comprehensive open-source framework based on *DBpedia* to extract lexical-semantic resources – an ontology about language use – from *Wiktionary*. The data currently includes language, part of speech, senses, definitions, synonyms, translations and taxonomies (hyponyms, hyperonyms, synonyms, antonyms) for each lexical word. Main focus is on flexibility to the loose schema and configurability towards differing language-editions of *Wiktionary*. This is achieved by a declarative mediator/wrapper approach. The goal is to allow the addition of languages just by configuration without the need of programming, thus enabling the swift and resource-conserving adaption of wrappers by domain experts. The extracted data is as fine granular as the source data in *Wiktionary* and additionally follows the *lemon* model. It enables use cases like disambiguation or machine translation. By offering a linked data service, we hope to extend DBpedia's central role in the LOD infrastructure to the world of Open Linguistics.

## 1 Introduction

The exploitation of community-built lexical resources has been discussed repeatedly. *Wiktionary* is one of the biggest collaboratively created lexical-semantic and linguistic resources available, written in 171 languages of which approximately 147 can be considered active[1], containing information about hundreds of spoken and even ancient languages. For example, the English *Wiktionary* contains nearly 3 million words[2]. A *Wiktionary* page provides for a lexical word a hierarchical disambiguation to its language, part of speech, sometimes etymologies and most prominently senses. Within this tree numerous kinds of linguistic properties are given, including synonyms, hyponyms, hyperonyms, example sentences, links to Wikipedia and many more. [13] gave a comprehensive overview on why

[*] Contributed equally to this work.

[1] http://s23.org/wikistats/wiktionaries_html.php

[2] See http://en.wiktionary.org/wiki/semantic for a simple example page.

H. Takeda et al. (Eds.): JIST 2012, LNCS 7774, pp. 191–206, 2013.

this dataset is so promising and how the extracted data can be automatically enriched and consolidated. Aside from building an upper-level ontology, one can use the data to improve NLP solutions, using it as comprehensive background knowledge. The noise should be lower when compared to other automatic generated text copora (e.g. by web crawling) as all information in *Wiktionary* is entered and curated by humans. Opposed to expert-built resources, the openness attracts a huge number of editors and thus enables a faster adaption to changes within the language.

The fast changing nature together with the fragmentation of the project into *Wiktionary language editions* (*WLE*) with independent layout rules, called *ELE guidelines* (Entry Layout Explained, see Section 3.2) poses the biggest problem to the automated transformation into a structured knowledge base. We identified this as a serious problem: Although the value of *Wiktionary* is known and usage scenarios are obvious, only some rudimentary tools exist to extract data from it. Either they focus on a specific subset of the data or they only cover one or two WLE. The development of a flexible and powerful tool is challenging to be accommodated in a mature software architecture and has been neglected in the past. Existing tools can be seen as adapters to single WLE — they are hard to maintain and there are too many languages, that constantly change. Each change in the *Wiktionary* layout requires a programmer to refactor complex code. The last years showed, that only a fraction of the available data is extracted and there is no comprehensive RDF dataset available yet. The key question is: Can the lessons learned by the successful DBpedia project be applied to *Wiktionary*, although it is fundamentally different from Wikipedia? The critical difference is that only word forms are formatted in infobox-like structures (e.g. tables). Most information is formatted covering the complete page with custom headings and often lists. Even the infoboxes itself are not easily extractable by default DBpedia mechanisms, because in contrast to DBpedias *one entity per page* paradigm, *Wiktionary* pages contain information about *several* entities forming a complex graph, i.e. the pages describe the lexical word, which occurs in several languages with different senses per part of speech and most properties are defined *in context* of such child entities. Opposed to the currently employed classic and straight-forward approach (implementing software adapters for scraping), we propose a declarative mediator/wrapper pattern. The aim is to enable non-programmers (the community of adopters and domain experts) to tailor and maintain the WLE wrappers themselves. We created a simple XML dialect to encode the ELE guidelines and declare triple patterns, that define how the resulting RDF should be built. This configuration is interpreted and run against *Wiktionary* dumps. The resulting dataset is open in every aspect and hosted as linked data[3]. Furthermore the presented approach can be extended easily to interpret or *triplify* other MediaWiki installations or even general document collections, if they follow a global layout.

[3] `http://wiktionary.dbpedia.org/`

**Table 1.** Comparison of existing Wiktionary approaches (ld = linked data hosting). None of the above include any crowd-sourcing approaches for data extraction. The wikokit dump is not in RDF.

| **name** | **active** | **available** | **RDF** | **#triples** | **ld** | **languages** |
|---|---|---|---|---|---|---|
| JWKTL | ✓ | dumps | ✗ | - | ✗ | en, de |
| wikokit | ✓ | source + dumps | ✓ | n/a | ✗ | en, ru |
| texai | ✗ | dumps | ✓ | ~ 2.7 million | ✗ | en |
| lemon scraper | ✓ | dumps | ✓ | ~16k per lang | ✗ | 6 |
| blexisma | ✗ | source | ✗ | - | ✗ | en |
| WISIGOTH | ✗ | dumps | ✗ | - | ✗ | en, fr |
| lexvo.org | ✓ | dumps | ✓ | ~353k | ✓ | en |

## 2 Related Work

In the last five years, the importance of *Wiktionary* as a lexical-semantic resource has been examined by multiple studies. Meyer et al. ([12,11]) presented an impressive overview on the importance and richness of *Wiktionary*. In [21] the authors presented the *JWKTL* framework to access *Wiktionary* dumps via a Java API. In [13] this JWKTL framework was used to construct an upper ontology called *OntoWiktionary*. The framework is reused within the *UBY project* [4], an effort to integrate multiple lexical resources (besides *Wiktionary* also *WordNet*, *GermaNet*, *OmegaWiki*, *FrameNet*, *VerbNet* and *Wikipedia*). The resulting dataset is modelled according to the *LMF ISO standard*[6]. [14] and [18] discussed the use of *Wiktionary* to canonicalize annotations on cultural heritage texts, namely the Thompson Motif-index. Zesch et. al. also showed, that *Wiktionary* is suitable for calculating semantic relatedness and synonym detection; and it outperforms classic approaches [22,20]. Furthermore, other NLP tasks such as sentiment analysis have been conducted with the help of *Wiktionary* [2]. Several questions arise, when evaluating the above approaches: Why are there not more NLP tools reusing the free *Wiktionary* data? Why are there no web mashups of the data[4]? Why has *Wiktionary* not become the central linking hub of lexical-semantic resources, yet?

From our point of view, the answer lies in the fact, that although the above papers presented various desirable properties and many use cases, they did not solve the underlying knowledge extraction and data integration task sufficiently in terms of coverage, precision and flexibility. Each of the approaches presented in Table 1 relies on tools to extract machine-readable data in the first place. In our opinion these tools should be seen independent from their respective usage and it is not our intention to comment on the scientific projects built upon them in any way here. We will show the state of the art and which open questions they raise.

[4] For example in an online dictionary from `http://en.wikipedia.org/wiki/List_of_online_dictionaries`

*JWKTL* is used as data backend of *OntoWiktionary* as well as UBY[5] and features a modular architecture, which allows the easy addition of new extractors (for example *wikokit* [8] is incorporated). The Java binaries and the data dumps in LMF are publicly available. Among other things, the dump also contains a mapping from concepts to lexicalizations as well as properties for part of speech, definitions, synonyms and subsumption relations. The available languages are English, German (both natively) and Russian through *wikokit*. According to our judgement, *JWKTL* can be considered the most mature approach regarding software architecture and coverage and is the current state of the art. *Texai*[6] and *Blexisma*[7] are also Java based APIs, but are not maintained anymore and were most probably made obsolete by changes to the *Wiktionary* layout since 2009. There is no documentation available regarding scope or intended granularity. A very fine grained extraction was conducted using WISIGOTH [17], but unfortunately there are no sources available and the project is unmaintained since 2010. Two newer approaches are the *lexvo.org* service and the algorithm presented in [9]. The *lexvo.org* service offers a linked data representation of *Wiktionary* with a limited granularity, namely it does not disambiguate on sense level. The source code is not available and only the English *Wiktionary* is parsed. As part of the Monnet project[8], McCrae et al. [9] presented a simple scraper to transform *Wiktionary* to the *lemon* RDF model [10]. The algorithm (like many others) makes assumptions about the used page schema and omits details about solving common difficulties as shown in the next section. At the point of writing, the sources are not available, but they are expected to be published in the future. Although this approach appears to be the state of the art regarding RDF modelling and linking, the described algorithm will *not scale to the community-driven heterogeneity* as to be defined in Section 3. All in all, there exist various tools that implement extraction approaches at various levels of granularity or output format. In the next section, we will show several challenges that in our opinion are insufficiently tackled by the presented approaches. Note that this claim is not meant to diminish the contribution of the other approaches as they were mostly created for solving a single research challenge instead of aiming to establish *Wiktionary* as a stable point of reference in computational linguistics using linked data.

## 3 Problem Description

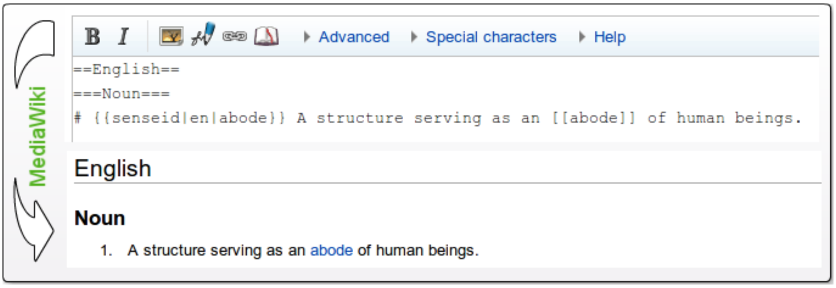
In order to conceive a flexible, effective and efficient solution, we survey in this section the challenges associated with Wiki syntax, *Wiktionary* and large-scale extraction.

[5] `http://www.ukp.tu-darmstadt.de/data/lexical-resources/uby/`, `http://www.ukp.tu-darmstadt.de/data/lexical-resources/uby/`

[6] `http://sourceforge.net/projects/texai/`

[7] `http://blexisma.ligforge.imag.fr/index.html`

[8] See `http://www.monnet-project.eu/`. A list of the adopted languages and dump files can be found at `http://monnetproject.deri.ie/lemonsource/Special:PublicLexica`



```
==English==
===Noun===
# {{senseid|en|abode}} A structure serving as an [[abode]] of human beings.
```

English

**Noun**

1. A structure serving as an abode of human beings.

**Fig. 1.** An excerpt of the *Wiktionary* page *house* with the rendered HTML

### 3.1 Processing Wiki Syntax

Pages in *Wiktionary* are formatted using the *wikitext* markup language[9]. Operating on the parsed HTML pages, rendered by the *MediaWiki engine*, does not provide any significant benefit, because the rendered HTML does not add any valuable information for extraction. Processing the database backup XML dumps[10] instead, is convenient as we could reuse the DBpedia extraction framework[11] in our implementation. The framework mainly provides input and output handling and also has built-in multi-threading by design. Actual features of the wikitext syntax are not notably relevant for the extraction approach, but we will give a brief introduction to the reader, to get familiar with the topic. A wiki page is formatted using the lightweight (easy to learn, quick to write) markup language *wikitext*. Upon request of a page, the MediaWiki engine renders this to an HTML page and sends it to the user's browser. An excerpt of the *Wiktionary* page *house* and the resulting rendered page are shown in Figure 1.

The markup == is used to denote headings, # denotes a numbered list with * for bullets, `[[link label]]` denotes links and `{{}}` calls a template. Templates are user-defined rendering functions that provide shortcuts aiming to simplify manual editing and ensuring consistency among similarly structured content elements. In MediaWiki, they are defined on special pages in the `Template:` namespace. Templates can contain any wikitext expansion, HTML rendering instructions and placeholders for arguments. In the example page in Figure 1, the `senseid` template[12] is used, which does nothing being visible on the rendered page, but adds an id attribute to the HTML `li`-tag, which is created by using `#`. If the English *Wiktionary* community decides to change the layout of senseid definitions at some point in the future , only a single change to the template definition is required. Templates are used heavily throughout *Wiktionary*, because

[9] `http://www.mediawiki.org/wiki/Markup_spec`

[10] `http://dumps.wikimedia.org/backup-index.html`

[11] `http://wiki.dbpedia.org/Documentation`

[12] `http://en.wiktionary.org/wiki/Template:senseid`

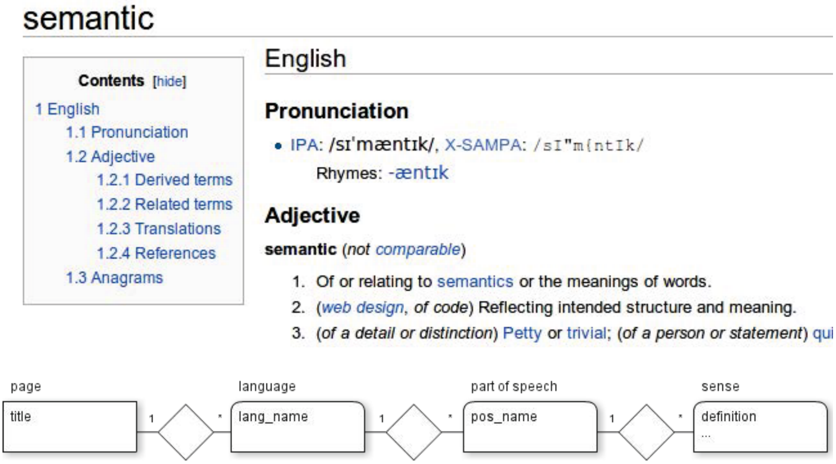
they substantially increase maintainability and consistency. But they also pose a problem to extraction: on the unparsed page only the template name and its arguments are available. Mostly this is sufficient, but if the template adds static information or conducts complex operations on the arguments, which is fortunately rare, the template result can only be obtained by a running MediaWiki installation hosting the pages. The resolution of template calls at extraction time slows the process down notably and adds additional uncertainty.

### 3.2 Wiktionary

*Wiktionary* has some unique and valuable properties:

- **Crowd-sourced**
  *Wiktionary* is community edited, instead of expert-built or automatically generated from text corpora. Depending on the activeness of its community, it is up-to-date to recent changes in the language, changing perspectives or new research. The editors are mostly semi-professionals (or guided by one) and enforce a strict editing policy. Vandalism is reverted quickly and bots support editors by fixing simple mistakes and adding automatically generated content. The community is smaller than Wikipedia's but still quite vital (between 50 and 80 very active editors with more than 100 edits per month for the English *Wiktionary* in 2012[13]).
- **Multilingual**
  The data is split into different Wiktionary Language Editions (WLE, one for each language). This enables the independent administration by communities and leaves the possibility to have different perspectives, focus and localization. Simultaneously one WLE describes multiple languages; only the representation language is restricted. For example, the German *Wiktionary* contains German description of German words **as well as** German descriptions for English, Spanish or Chinese words. Particularly the linking across languages shapes the unique value of *Wiktionary* as a rich multi-lingual linguistic resource. Especially the WLE for not widely spread languages are valuable, as corpora might be rare and experts are hard to find.
- **Feature rich**
  As stated before, *Wiktionary* contains for each lexical word –A lexical word is just a string of characters and has no disambiguated meaning yet– a disambiguation regarding language, part of speech, etymology and senses. Numerous additional linguistic properties exist normally for each part of speech. Such properties include word forms, taxonomies (hyponyms, hyperonyms, synonyms, antonyms) and translations. Well maintained pages (e.g. frequent words) often have more sophisticated properties such as derived terms, related terms and anagrams.

[13] `http://stats.wikimedia.org/wiktionary/EN/TablesWikipediaEN.htm`



semantic

Contents [hide]

1 English
1.1 Pronunciation
1.2 Adjective
1.2.1 Derived terms
1.2.2 Related terms
1.2.3 Translations
1.2.4 References
1.3 Anagrams

English

Pronunciation

- IPA: /sɪˈmæntɪk/, X-SAMPA: /sI"m{ntIk/
  Rhymes: -æntɪk

Adjective

**semantic** (*not comparable*)

1. Of or relating to semantics or the meanings of words.
2. (*web design*, *of code*) Reflecting intended structure and meaning.
3. (*of a detail or distinction*) Petty or trivial; (*of a person or statement*) qui

**Fig. 2.** Example page `http://en.wiktionary.org/wiki/semantic` and underlying schema, only valid for the English *Wiktionary*, as other WLE might look very different

- **Open license**
  All the content is dual-licensed under both the *Creative Commons CC-BY-SA 3.0 Unported License*[14] as well as the *GNU Free Documentation License (GFDL)*.[15] All the data extracted by our approach falls under the same licences.
- **Big and growing**
  English contains 2,9M pages, French 2,1M, Chinese 1,2M, German 0,2 M. The overall size (12M pages) of *Wiktionary* is in the same order of magnitude as Wikipedia's size (20M pages)[16]. The number of edits per month in the English *Wiktionary* varies between 100k and 1M — with an average of 200k for 2012 so far. The number of pages grows — in the English *Wiktionary* with approx. 1k per day in 2012.[17]

The most important resource to understand how *Wiktionary* is organized are the *Entry Layout Explained* (ELE) help pages. As described above, a page is divided into sections that separate languages, part of speech etc. The table of content on the top of each page also gives an overview of the hierarchical structure. This hierarchy is already very valuable as it can be used to disambiguate a lexical word. The schema for this tree is restricted by the ELE guidelines[18]. The entities illustrated in Figure 3.2 of the ER diagram will be called *block* from now on. The schema can differ between WLEs and normally evolves over time.

[14] `http://en.wiktionary.org/wiki/Wiktionary:Text_of_Creative_Commons_Attribution-ShareAlike_3.0_Unported_License`

[15] `http://en.wiktionary.org/wiki/Wiktionary:GNU_Free_Documentation_License`

[16] `http://meta.wikimedia.org/wiki/Template:Wikimedia_Growth`

[17] `http://stats.wikimedia.org/wiktionary/EN/TablesWikipediaEN.htm`

[18] For English see `http://en.wiktionary.org/wiki/Wiktionary:ELE`

### 3.3 Wiki-Scale Data Extraction

The above listed properties that make *Wiktionary* so valuable, unfortunately pose a serious challenge to extraction and data integration efforts. Conducting an extraction for specific languages at a fixed point in time is indeed easy, but it eliminates some of the main features of the source. To fully synchronize a knowledge base with a community-driven source, one needs to make distinct design choices to fully capture all desired benefits. MediaWiki was designed to appeal to non-technical editors and abstains from intensive error checking as well as formally following a grammar — the community gives itself just layout guidelines. One will encounter fuzzy modelling and unexpected information. Editors often see no problem with such "noise" as long as the page's visual rendering is acceptable. Overall, the main challenges can be summed up as (1) the constant and frequent changes to data *and schema*, (2) the heterogeneity in WLE schemas and (3) the human-centric nature of a wiki.

## 4 Design and Implementation

Existing extractors as presented in Section 2 mostly suffer from their *inflexible* nature resulting from their narrow use cases at development time. Very often approaches were only implemented to accomplish a short term goal (e.g. prove a scientific claim) and only the needed data was extracted in an *ad-hoc* manner. Such evolutionary development generally makes it difficult to generalize the implementation to heterogeneous schemas of different WLE. Most importantly, however, they ignore the community nature of a *Wiktionary*. Fast changes of the data require ongoing maintenance, ideally by the wiki editors from the community itself or at least in tight collaboration with them. These circumstances pose serious requirements to software design choices and should not be neglected. All existing tools are rather monolithic, hard-coded black boxes. Implementing a new WLE or making a major change in the WLE's ELE guidelines will require a programmer to refactor most of its application logic. Even small changes like new properties or naming conventions will require software engineers to align settings. The amount of maintenance work necessary for the extraction correlates with change frequency in the source. Following this argumentation, a community-built resource can only be efficiently extracted by a community-configured extractor. This argument is supported by the successful crowd-sourcing of DBpedia's internationalization [7] and the non-existence of *open* alternatives with equal extensiveness.

Given these findings, we can now conclude four high-level requirements:

- declarative description of the page schema;
- declarative information/token extraction, using a terse syntax, maintainable by non-programmers;
- configurable mapping from language-specific tokens to a global vocabulary;
- fault tolerance (uninterpretable data is skipped).

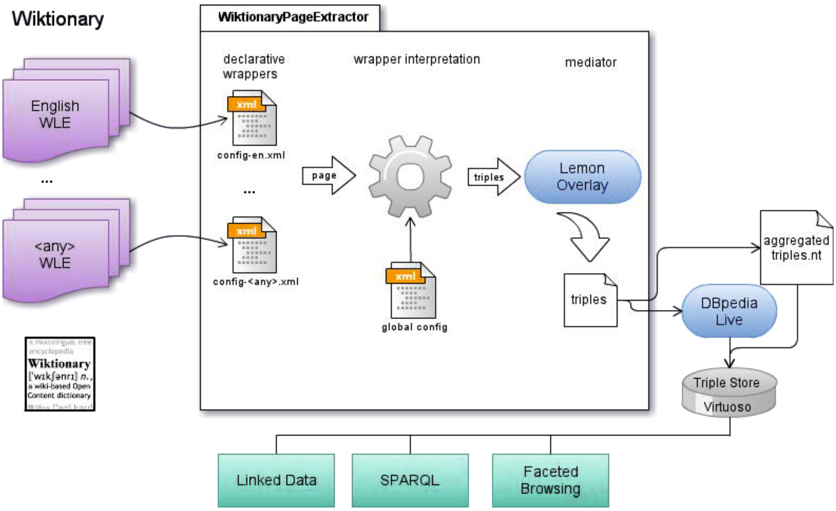


**Fig. 3.** Architecture for extracting semantics from Wiktionary leveraging the DBpedia framework

We solve the above requirements by proposing an extension to the DBpedia framework (in fact an additional extractor), which follows a rather sophisticated workflow, shown in Figure 3.

The *Wiktionary* extractor is invoked by the DBpedia framework to handle a page. It therefore uses a language-specific configuration file, that has to be tailored to match the WLE's ELE guidelines to interpret the page. At first, the resulting triples still adhere to a language-specific schema, that directly reflects the assumed layout of the WLE. A generic lossless transformation and annotation using the *lemon* vocabulary is then applied to enforce a global schema and reduce semantic heterogeneity. Afterwards the triples are returned to the DBpedia frameworks, which takes care of the serialization and (optionally) the synchronization with a triple store via DBpedia Live[19]. The process of interpreting the declarative wrapper is explained in more detailed in Figure 4.

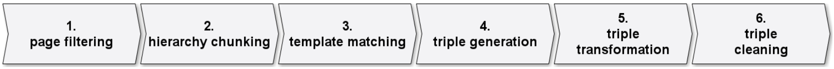


**Fig. 4.** Overview of the extractor workflow

[19] http://live.dbpedia.org/live

### 4.1 Extraction Templates

As mentioned in Section 3.2, we define *block* as the part of the hierarchical page that is responsible for a certain entity in the extracted RDF graph. For each *block*, there can be declarations on how to process the page on that level. This is done by so called *extraction templates*(ET) (not to be confused with the templates of *wikitext*). Each possible section in the *Wiktionary* page layout (i.e. each linguistic property) has an ET configured (explained in detail below). The idea is to provide a declarative and intuitive way to encode *what to extract*. For example consider the following page snippet:

```
===Synonyms===
* [[building]]
* [[company]]
```

Since the goal is to emit a link to each resource per line, we can write the ET in the following style, using the popular scraping paradigms such as regular expressions:

```
===Synonyms===
(* [[\$target]]
)+
```

Some simple constructs for variables "`$target`" and loops "`(*`", "`)+`" are defined for the ET syntax. If they are *matched against* an actual wiki page, *bindings* are extracted by a matching algorithm. We omit a low-level, technical description of the algorithm — one can think of it like a Regular Expression *Named Capturing Group*. The found *variable bindings* for the above example are `{(target->building), (target->company)}`. The triple generation rule encoded in XML looks like:

```
<triple s="http://some.ns/$entityId" p="http://some.ns/hasSynonym" o="http://some.ns/$target" />
```

Notice the reuse of the `$target` variable: The data extracted from the page is inserted into a triple. The variable `$entityId` is a reserved global variable, that holds the page name e.g. the word. The created triples in N-Triples syntax are:

```
<http://some.ns/house>   <http://some.ns/hasSynonym>   <http://some.ns/building> .
<http://some.ns/house>   <http://some.ns/hasSynonym>   <http://some.ns/company> .
```

The actual patterns are more complex, but the mechanism is consistently used throughout the system.

### 4.2 Algorithm

The algorithm of processing a page works as follows:

*Input:* Parsed page obtained from the DBpedia Framework (essentially a lexer is used to split the Wiki Syntax into tokens)

1. Filter irrelevant pages (user/admin pages, statistics, list of things, files, templates, etc.) by applying string comparisons on the page title. Return an empty result on that condition.

2. Build a finite state automaton[20] from the page layout encoded in the WLE specific XML configuration. This schema also contains so called *indicator templates* for each *block*, that — if they match at the current page token — indicate that their respective block starts. So they trigger state transitions. In this respect the mechanism is similar to [9], but in contrast our approach is declarative — the automaton is constructed *on-the-fly* and not hard-coded. The current state represents the current position in the disambiguation tree.
3. The page is processed token by token:
   (a) Check if *indicator templates* match. If yes, the corresponding block is entered. The *indicator templates* also emit triples like in the *extraction template* step below. These triples represent the block in RDF – for example the resource `http://wiktionary.dbpedia.org/resource/semantic-English` represents the English block of the page "semantic".
   (b) Check if any *extraction template* of the current block match.
   – If yes, transform the variable bindings to triples.[21] Localization specific tokens are replaced as configured in the so called *language mapping* (explained in detail in section 4.3).
4. The triples are then *transformed*. In our implementation *transformation* means, that all triples are handed to a static function, which return a set of triples again. One could easily load the triples into a triple store like JENA and apply arbitrary SPARQL Construct and Update transformations. This step basically allows post-processing, e.g. consolidation, enrichment or annotation. In our case, we apply the schema transformation (by the mediator) explained in detail in Section 4.4).
5. The triples are sorted and de-duplicated to remove redundancy in the RDF dumps.

*Output:* Set of triples (handed back to the DBpedia Framework).

### 4.3 Language Mapping

The language mappings are a very simple way to translate and normalize tokens, that appear in a WLE. In the German WLE, for example, a noun is described with the German word "*Substantiv*". Those tokens are translated to a shared vocabulary, before emitting them (as URIs for example). The configuration is also done within the language specific XML configuration:

```
<mapping from="Substantiv" to="Noun">
<mapping from="Deutsch" to="German">
...
```

[20] Actually a finite state transducer, most similar to the Mealy-Model.

[21] In our implementation: Either declarative rules are given in the XML config or alternatively static methods are invoked on user-defined classes (implementing a special interface) for an imperative transformation. This can greatly simplify the writing of complex transformation.

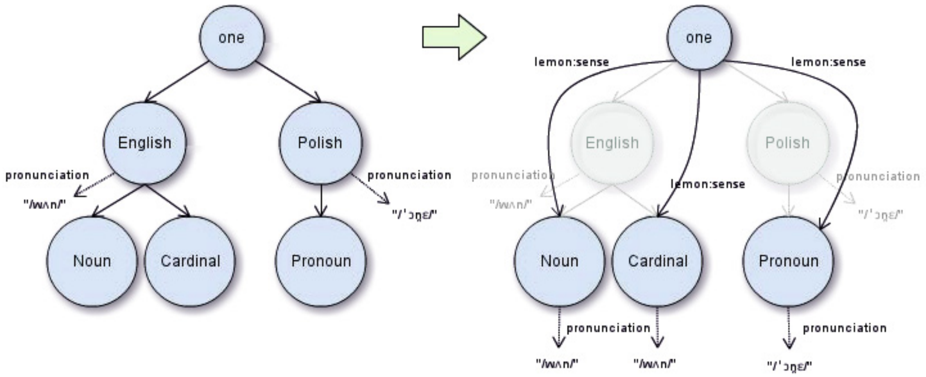


**Fig. 5.** Schema normalization

### 4.4 Schema Mediation by Annotation with *lemon*

The last step of the data integration process is the schema normalization. The global schema of all WLE is not constructed in a centralized fashion — instead we found a way to both making the data globally navigable and keeping the heterogeneous schema without loosing information. *lemon* [10] is an RDF model for representing lexical information (with links to ontologies — possibly DBpedia). We use part of that model to encode the relation between *lexical entries* and *lexical senses*. *lemon* has great potential of becoming the *de facto* standard for representing dictionaries and lexica in RDF and is currently the topic of the OntoLex W3C Community group[22]. The rationale is to add *shortcuts* from *lexical entities* to *senses* and propagate properties that are along the intermediate nodes down to the senses. This can be accomplished with a generic algorithm (a generic tree transformation, regardless of the depth of the tree and used links). Applications assuming only a *lemon* model, can operate on the shortcuts and if applied as an overlay — leaving the original tree intact — this still allows applications, to also operate on the actual tree layout. The (simplified) procedure is presented in Figure 5[23]. The use of the *lemon* vocabulary and model as an additional schema layer can be seen as our mediator. This approach is both lightweight and effective as it takes advantage of *multi-schema modelling*.

## 5 Resulting Data

The extraction has been conducted as a proof-of-concept on four major WLE: The English, French, German and Russian *Wiktionary*. The datasets combined

[22] `http://www.w3.org/community/ontolex/`

[23] Note, that in the illustration it could seem like the information about part-of-speech would be missing in the *lemon* model. This in not the case. Actually from the part-of-speech nodes, there is a link to corresponding language nodes. These links are also propagated down the tree.

**Table 2.** Statistical comparison of extractions for different languages. XML lines measures the number of lines of the XML configuration files.

| language | *#words* | *#triples* | *#resources* | *#predicates* | *#senses* | *XML lines* |
|---|---|---|---|---|---|---|
| *en* | 2,142,237 | 28,593,364 | 11,804,039 | 28 | 424,386 | 930 |
| *fr* | 4,657,817 | 35,032,121 | 20,462,349 | 22 | 592,351 | 490 |
| *ru* | 1,080,156 | 12,813,437 | 5,994,560 | 17 | 149,859 | 1449 |
| *de* | 701,739 | 5,618,508 | 2,966,867 | 16 | 122,362 | 671 |

**Table 3.** Statistical quality comparison

| language | *t/w* | *#wws* | *s/wws* | *t/l* |
|---|---|---|---|---|
| *en* | 13.35 | 591,073 | 1.39 | 2.70 |
| *fr* | 7.52 | 750,206 | 1.26 | 1.73 |
| *ru* | 11.86 | 211,195 | 1.40 | 2.25 |
| *de* | 8.01 | 176,122 | 1.43 | 1.06 |

contain more than 80 million facts. The data is available as N-Triples dumps[24], Linked Data[25], via the *Virtuoso Faceted Browser*[26] or a SPARQL endpoint[27]. Table 2 compares the size of the datasets from a quantitative perspective.

The statistics show, that the extraction produces a vast amount of data with broad coverage, thus resulting in the largest lexical linked data resource. There might be partially data quality issues with regard to missing information (for example the number of *words with senses* seems to be relatively low intuitively), but detailed quality analysis has yet to be done. Instead we defined some simple quality measures that can be automatically computed.

Table 3 gives an assessment of the quality of the language configuration independent from the quality of the underlying source data:
*t/w*: *Triples per word.* The simplest measure of information density. *#wws*: *Words with senses.* The number of words, that have at least one sense extracted. An indicator for the ratio of pages for which valuable information could be extracted, but consider stub pages, that are actually empty. *s/wws*: *Senses per word with sense.* Gives an idea of the average senses per word while ignoring unmaintained pages. *t/l*: *Triples per line.* The number of triples divided by the number of line breaks in the page source (plus one). Averaged across all pages.

## 6 Lessons Learned

*Making unstructured sources machine-readable creates feedback loops.* Although this is not yet proven by empirical data, the argument that extracting structured data from an open data source and making it freely available in turn encourages users of the extracted data to contribute to the source, seems reasonable. The clear

[24] `http://downloads.dbpedia.org/wiktionary`

[25] For example `http://wiktionary.dbpedia.org/resource/dog`

[26] `http://wiktionary.dbpedia.org/fct`

[27] `http://wiktionary.dbpedia.org/sparql`

incentive is to *get the data out again.* This increase in participation besides improving the source, also illustrates the advantages of machine readable data to common Wiktionarians. Such a positive effect from DBpedia supported the current *Wikidata*[28] project.

*Suggested changes to Wiktionary.* Although it's hard to persuade the community of far-reaching changes, we want to conclude how *Wiktionary* can increase its data quality and enable better extraction.

- **Homogenize Entry Layout across all WLE's.**
- **Use anchors to markup senses:** This implies creating URIs for senses. These can then be used to be more specific when referencing a *word* from another article. This would greatly benefit the evaluation of automatic anchoring approaches like in [13].
- **Word forms:** The notion of word forms (e.g. declensions or conjugations) is not consistent across articles. They are hard to extract and often not given.

## 7 Discussion and Future Work

Our main contributions are an extremely flexible extraction from *Wiktionary*, with simple adaption to new Wiktionaries and changes via a declarative configuration. By doing so, we are provisioning a linguistic knowledge base with unprecedented detail and coverage. The DBpedia project provides a mature, reusable infrastructure including a public Linked Data service and SPARQL endpoint. All resources related to our *Wiktionary* extraction, such as source-code, extraction results, pointers to applications etc. are available from our project page.[29] As a result, we hope it will evolve into a central resource and interlinking hub on the currently emerging Web of Linguistic Data.

### 7.1 Next Steps

**Wiktionary Live:** Users constantly revise articles. Hence, data can quickly become outdated, and articles need to be re-extracted. DBpedia-Live enables such a continuous synchronization between DBpedia and Wikipedia. The WikiMedia foundation kindly provided us access to their update stream, the Wikipedia OAI-PMH[30] live feed. The approach is equally applicable to *Wiktionary*. The *Wiktionary* Live extraction will enable users for the first time ever to query *Wiktionary* like a database in real-time and receive up-to-date data in a machine-readable format. This will strengthen *Wiktionary* as a central resource and allow it to extend its coverage and quality even more.

**Wiki Based UI for the WLE Configurations:** To enable the crowd-sourcing of the extractor configuration, an intuitive web interface is desirable. Analogue to the mappings wiki[31] of DBpedia, a wiki could help to hide the technical details of the

[28] `http://meta.wikimedia.org/wiki/Wikidata`

[29] `http://wiktionary.dbpedia.org`

[30] Open Archives Initiative Protocol for Metadata Harvesting, cf. `http://www.mediawiki.org/wiki/Extension:OAIRepository`

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

configuration even more. Therefore a JavaScript based WYSIWYG XML editor seems useful. There are various implementations, which can be easily adapted.

**Linking:** Finally, an alignment with existing linguistic resources like WordNet and general ontologies like YAGO or DBpedia is essential. That way *Wiktionary* will allow for the interoperability across a multilingual semantic web.

### 7.2 Open Research Questions

**Publishing Lexica as Linked Data.** The need to publish lexical resources as linked data has been recognized recently [16]. Although principles for publishing RDF as Linked Data are already well established [1], the choice of identifiers and first-class objects is crucial for any linking approach. A number of questions need to be clarified, such as which entities in the lexicon can be linked to others. Obvious candidates are entries, senses, synsets, lexical forms, languages, ontology instances and classes, but different levels of granularity have to be considered and a standard linking relation such as `owl:sameAs` will not be sufficient. Linking across data sources is at the heart of linked data. An open question is how lexical resources with differing schemata can be linked and how are linguistic entities to be linked with ontological ones. There is most certainly an impedance mismatch to bridge.

The success of DBpedia as a “crystallization point for the Web of Data” is predicated on the stable identifiers provided by Wikipedia and are an obvious prerequisite for any data authority. Our approach has the potential to drive this process by providing best practices and live showcases and data in the same way DBpedia has provided it for the LOD cloud. Especially, our work has to be seen in the context of the recently published Linguistic Linked Data Cloud[3] and the community effort around the Open Linguistics Working Group (OWLG)[32] and NIF [5]. Our Wiktionary conversion project provides valuable data dumps and linked data services to further fuel development in this area.

**Algorithms and Methods to Bootstrap and Maintain a Lexical Linked Data Web.** State-of-the-art approaches for interlinking instances in RDF knowledge bases are mainly build upon similarity metrics [15,19] to find duplicates in the data, linkable via `owl:sameAs`. Such approaches are not directly applicable to lexical data. Existing linking properties either carry strong formal implications (e.g. `owl:sameAs`) or do not carry sufficient domain-specific information for modelling semantic relations between lexical knowledge bases.

## References

1. Auer, S., Lehmann, J.: Making the web a data washing machine - creating knowledge out of interlinked data. Semantic Web Journal (2010)
2. Chesley, P., Vincent, B., Xu, L., Srihari, R.K.: Using verbs and adjectives to automatically classify blog sentiment. In: AAAI Spring Symposium (2006)
3. Chiarcos, C., Hellmann, S., Nordhoff, S., Moran, S., Littauer, R., Eckle-Kohler, J., Gurevych, I., Hartmann, S., Matuschek, M., Meyer, C.M.: The open linguistics working group. In: LREC (2012)

[32] `http://linguistics.okfn.org`

4. Gurevych, I., Eckle-Kohler, J., Hartmann, S., Matuschek, M., Meyer, C.M., Wirth, C.: Uby - a large-scale unified lexical-semantic resource based on lmf. In: EACL 2012 (2012)
5. Hellmann, S., Lehmann, J., Auer, S.: Linked-data aware URI schemes for referencing text fragments. In: ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (eds.) EKAW 2012. LNCS, vol. 7603, pp. 175–184. Springer, Heidelberg (2012)
6. ISO 24613:2008. Language resource management – Lexical markup framework. ISO, Geneva, Switzerland
7. Kontokostas, D., Bratsas, C., Auer, S., Hellmann, S., Antoniou, I., Metakides, G.: Internationalization of Linked Data: The case of the Greek DBpedia edition. Journal of Web Semantics (2012)
8. Krizhanovsky, A.A.: Transformation of wiktionary entry structure into tables and relations in a relational database schema. CoRR (2010), `http://arxiv.org/abs/1011.1368`
9. McCrae, J., Cimiano, P., Montiel-Ponsoda, E.: Integrating WordNet and Wiktionary with lemon. In: Chiarcos, C., Nordhoff, S., Hellmann, S. (eds.) Linked Data in Linguistics. Springer (2012)
10. McCrae, J., Spohr, D., Cimiano, P.: Linking Lexical Resources and Ontologies on the Semantic Web with Lemon. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 245–259. Springer, Heidelberg (2011)
11. Meyer, C.M., Gurevych, I.: How web communities analyze human language: Word senses in wiktionary. In: Second Web Science Conference (2010)
12. Meyer, C.M., Gurevych, I.: Worth its weight in gold or yet another resource — A comparative study of wiktionary, openThesaurus and germaNet. In: Gelbukh, A. (ed.) CICLing 2010. LNCS, vol. 6008, pp. 38–49. Springer, Heidelberg (2010)
13. Meyer, C.M., Gurevych, I.: OntoWiktionary – Constructing an Ontology from the Collaborative Online Dictionary Wiktionary. In: Semi-Automatic Ontology Development: Processes and Resources. IGI Global (2011)
14. Moerth, K., Declerck, T., Lendvai, P., Váradi, T.: Accessing multilingual data on the web for the semantic annotation of cultural heritage texts. In: 2nd Workshop on the MSW, ISWC (2011)
15. Ngonga Ngomo, A.-C., Auer, S.: Limes - a time-efficient approach for large-scale link discovery on the web of data. In: Proceedings of IJCAI (2011)
16. Nuzzolese, A.G., Gangemi, A., Presutti, V.: Gathering lexical linked data and knowledge patterns from framenet. In: K-CAP (2011)
17. Sajous, F., Navarro, E., Gaume, B., Prévot, L., Chudy, Y.: Semi-automatic Endogenous Enrichment of Collaboratively Constructed Lexical Resources: Piggybacking onto Wiktionary. In: Loftsson, H., Rögnvaldsson, E., Helgadóttir, S. (eds.) IceTAL 2010. LNCS (LNAI), vol. 6233, pp. 332–344. Springer, Heidelberg (2010)
18. Mörth, K., Budin, G., Declerck, T., Lendvai, P., Váradi, T.: Towards linked language data for digital humanities
19. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Discovering and maintaining links on the web of data. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 650–665. Springer, Heidelberg (2009)
20. Weale, T., Brew, C., Fosler-Lussier, E.: Using the wiktionary graph structure for synonym detection. In: The People's Web Meets NLP, ACL-IJCNLP (2009)
21. Zesch, T., Müller, C., Gurevych, I.: Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In: LREC (2008)
22. Zesch, T., Müller, C., Gurevych, I.: Using wiktionary for computing semantic relatedness. In: AAAI (2008)