Leveraging DBnary Data to Enrich Information of Multiword Expressions in Wiktionary

Gilles Serasset, Thierry Declerck, Lenka Bajčetić

Leveraging DBnary Data to Enrich Information of Multiword Expressions in Wiktionary

Gilles Sérasset
Université Grenoble Alpes
CNRS, Grenoble INP*, LIG
38000 Grenoble, France
gilles.serasset@imag.fr

Thierry Declerck
DFKI GmbH, Multilingual Technologies
Saarland Informatics Campus D3 2
D-66123 Saarbrücken, Germany
declerck@dfki.de

Lenka Bajčetić
Innovation Center of the School of
Electrical Engineering in Belgrade
Bulevar kralja Aleksandra 73
11000 Belgrade, Serbia
lenka.bajcetic@ic.etf.ac.bg.rs

Abstract

We describe first an approach consisting of computing pronunciation information for multiword expressions (MWEs) included in the English edition of Wiktionary. During this work, we learnt about the DBnary resource, which represents information extracted from 23 language editions of Wiktionary in a Linked Open Data (LOD) compliant way. This lead to updates of the DBnary programs, to support the extraction of the desired pronunciation information for MWEs and which we document in this paper. The use by DBnary of LOD compliant models and vocabularies, more specifically of the *OntoLex-Lemon* model, opens the possibility for additional lexicographic enrichment of the MWEs, like adding morphosyntactic and semantic information to their components. DBnary is thus now more than "just" an extractor and mapper of Wiktionary data in a LOD representation, but is also contributing to the lexicographic enrichment of Wiktionary pages dealing with MWEs. In the longer term, our work will allow for more data on English MWEs to be made available in the Linguistic Linked Data cloud.

1 Introduction

Recent work (Bajčetić et al., 2023) dealing with the computation of pronunciation information for multiword expressions (MWEs) in the English edition of Wiktionary was using a combination of the Wikimedia API[1] to find wiki pages describing MWEs and of an XML parser to analyse and extract information from the corresponding wiki text.[2] This approach proved to be tedious and time-consuming. We decided therefore to use the DBnary resource, which is already providing for a structured representation of Wiktionary content, to get access to the Wiktionary data necessary for the computation of pronunciation information for MWEs and for exploring other tasks, like specifying the part-of-speech of components of MWEs or for associating semantic information to those components.

DBnary is a lexical resource extracted from 23 language editions of Wiktionary. Lexical data is represented using the Linked Open Data (LOD) principles[3] and as such it is using RDF[4] as its representation model. It is freely available and may be either downloaded or directly queried on the internet. DBnary uses the *OntoLex-Lemon* standard vocabulary (Cimiano et al., 2016),[5] displayed in Figure 1 to represent the lexical entries structures, along with *lexvo* (de Melo, 2015) to uniquely identify languages, *lexinfo* (Cimiano et al., 2011)[6] and *Olia* (Chiarcos and Sukhareva, 2015)[7] for linguis-

[1] https://en.wiktionary.org/w/api.php.

[2] One can also apply an XML parser to the full Wiktionary dump in XML format, available at https://dumps.wikimedia.org/enwiktionary/20230320/.

[3] See https://www.w3.org/wiki/LinkedData for more information on those principles.

[4] The Resource Description Framework (RDF) model is a graph based model for the representation of data and metadata, using URIs to represent resources (nodes) and properties (edges). See https://www.w3.org/TR/rdf11-primer/ for more details.

[5] See also the specification document at https://www.w3.org/2016/05/ontolex/.

[6] The latest version of the *lexinfo* ontology can be downloaded at https://lexinfo.net/.

[7] The "Ontologies of Linguistic Annotation (OLiA)" is available at https://acoli-repo.github.io/olia/.

tic data categories.

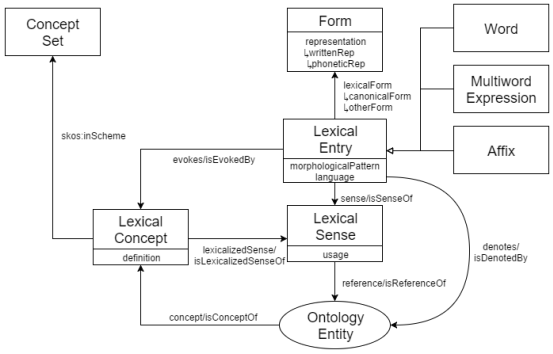

Figure 1: The core module OntoLex-Lemon. Taken from https://www.w3.org/2016/05/ontolex/#core

While trying to reproduce (Bajčetić et al., 2023) work, we noticed that DBnary was lacking some information. First, Wiktionary MWEs were not marked explicitly. Second, derivation relations between single word lexical entries and MWEs, in which they occur, were not extracted, while this information is crucial for the disambiguation of components of MWEs that are heteronyms (see Section 2 for a detailed discussion). The DBnary maintainer[8] tuned the extraction program to fix these identified lacks.

This paper summarises first the work presented in (Bajčetić et al., 2023) (section 2), providing details on the different means we used to access Wiktionary data (section 3), initially through API queries and XML parsing and finally using the latest version of DBnary for which we detail how we query it for accessing the necessary Wiktionary data. Section 4 presents and evaluates the computing of pronunciation information to be associated with Wiktionary MWEs. Then, in section 5, we discuss the promising use of the decomposition module of OntoLex-Lemon for supporting an enriched semantic representation of the components of MWEs.

2 Adding pronunciation information to multiword expressions in Wiktionary

In this section, we summarize the approach described in (Bajčetić et al., 2023), motivating also the decision to use DBnary as the primary source for the task of adding pronunciation information to Wiktionary MWEs, a move that lead to the fine-tuning of the extraction engine that is generating DBnary.

2.1 Wiktionary

Wiktionary[9] is a freely available web-based multilingual dictionary. Like other Wikimedia[10] supported initiatives, it is a collaborative project that is also integrating information from expert-based dictionary resources, when their licensing conditions allow it.

Wiktionary includes a thesaurus, a rhyme guide, phrase books, language statistics and extensive appendices. Wiktionary's information also (partly) includes etymologies, pronunciations, sample quotations, synonyms, antonyms and translations.[11] Wiktionary has also developed categorization practices, which classify an entry along the lines of linguistics (for example "developed terms by language") but also topical information (for example "en:Percoid fish").[12]

2.2 Multiword expressions in Wiktionary

Wiktionary introduces the category "English multiword terms" (MWT), which is defined as "lemmas that are an idiomatic combination of multiple words"[13], while Wiktionary has the page "multiword expression", categorized as a MWT and defined as "lexeme-like unit made up of a sequence of two or more words that has properties that are not predictable from the properties of the individual words or their normal mode of combination".[14] We see these two definitions are interchangeable, since they both focus on the aspect of non-compositionality of a lexeme built from multiple words. For consistency with common usage in NLP publications, we use in this paper the term

[8] The DBnary extraction programs are open source and available at: https://gitlab.com/gilles.serasset/dbnary/ where issues can be added to ask for correction or enhancement of the extractors. It is also possible to fix the extractors and create a Merge Request.

[9] https://en.wiktionary.org/

[10] https://www.wikimedia.org/

[11] See https://en.wikipedia.org/wiki/Wiktionary for more details.

[12] The entry "sea bass", for example, is categorized, among others, both as an instance of "English multiword terms" and of "en:Percoid fish". The categorization system is described at https://en.wiktionary.org/wiki/Wiktionary:Categorization

[13] https://en.wiktionary.org/wiki/Category:English_multiword_terms. This category is an instance of the umbrella category "Multiword terms by language", see https://en.wiktionary.org/wiki/Category:Multiword_terms_by_language.

[14] https://en.wiktionary.org/wiki/multi-word_expression.

Multiword Expression (MWE), but stress that they are categorized as MWTs in Wiktionary.

According to Wiktionary website, the current version of the English edition of Wiktionary is listing 157,753 pages containing an English MWE[15], and 75,389 pages containing an English term equipped with IPA pronunciation"[16]. This is quite a small number in comparison to the whole English Wiktionary, which has over 8,597,416 pages (with 7,365,114 items marked as "content pages", totalizing 226,078,477 words (https://en.wiktionary.org/wiki/Special:Statistics, [accessed 25.03.2023]). It is important to keep in mind that the English Wiktionary contains a lot of terms which are not English. We can see the exact number of English lemmas if we look at the Wiktionary category "English lemmas".[17] The actual number of 711,294 pages containing an English lemma means that a little over 10% of English lemmas have pronunciation, while approximately 22% of all English lemmas belong in the MWT category. So there is clearly a gap that needs to be filled when it comes to pronunciation information in Wiktionary. While introducing pronunciation for the remaining 90% of lemmas seems like it has to be a manual task (or semi-automatic, using another resource) - we have investigated ways to produce the missing pronunciation for numerous MWEs.

2.3 Overview of the approach for adding pronunciation information to MWEs

Bajčetić et al. (2023) describes the approach aiming at enriching English MWEs included in Wiktionary by pronunciation information extracted from their sub-parts. This endeavour itself is a continuation of work consisting of extracting pronunciation information from Wiktionary in order to enrich the Open English WordNet (McCrae et al., 2020),[18] where pronunciation information has been added only for single word entries, as described in (Declerck and Bajčetić, 2021).

An issue to deal with in this approach is the treatment of heteronyms that are a component of a MWE[19]. In order to select the correct pronunciation, an additional analysis of the Wiktionary data is needed, disambiguating between the different senses of the heteronym. This issue is multiplied by the number of MWEs containing such a heteronym. An example of such a case is given by the Wiktionary page "acoustic bass", for which our algorithm has to specify that the pronunciation /beɪs/ (and not /bæs/) has to be selected and combined with /əˈkuː.stɪk/.[20]

Since we need to semantically disambiguate one or more components of a MWE for generating its pronunciation, our work can lead to the addition of morphosyntactic and semantic information of those components and thus enrich the overall representation of the MWEs entries, a task we started to work on, and for which we consulted DBnary, and this step was leading to the development of a new version of the DBnary extractor, in order to explicitly mark MWEs and Wiktionary "derived terms", which establish semantic links between single word entries and MWEs in which they occur.

In order to implement our approach, we need thus to extract from Wiktionary:

- all existing pronunciation of English terms
- a list of all MWEs that are available
- all derivation relations between single English terms and their derived terms, when those are MWEs.

3 Accessing Wiktionary data

When it comes to extracting information from Wiktionary, we can usually find three approaches in the literature. Mainly, parsing the dumps, accessing Wktionary APIs or querying DBnary.

3.1 Parsing Wiktionary dumps

The first approach requires downloading the English Wiktionary dump and parsing it. The dump is an XML document containing the MediaWiki

[15] https://en.wiktionary.org/wiki/Category:English_multiword_terms, [accessed on the 25.03.2023]

[16] https://en.wiktionary.org/wiki/Category:English_terms_with_IPA_pronunciation

[17] https://en.wiktionary.org/wiki/Category:English_lemmas

[18] See also https://en-word.net/

[19] The online Oxford Dictionary gives this definition: "A heteronym is one of two or more words that have the same spelling but different meanings and pronunciation, for example 'tear' meaning 'rip' and 'tear' meaning 'liquid from the eye'", https://www.oxfordlearnersdictionaries.com/definition/english/heteronym

[20] The corresponding entry "bass" (the one marked with "Etymology 1") in the Wiktionary page https://en.wiktionary.org/wiki/bass#English lists 65 derived terms (most of them MWEs, and with only nine terms being equipped with pronunciation information), for which we can assume that the pronunciation /bæs/ has to be selected for the component "bass".

source (see Figure 2) of all entries and templates or modules defined in the English edition. Indeed, each entry is a kind of program whose execution results in the HTML page that is visible in your browser (see Figure 3).

```
====Pronunciation====
* {{enPR|bās}}, {{IPA|en|/beɪs/}}
* {{audio|en|en-us-bass-low.ogg|Audio (US)}}
* {{rhymes|en|eɪs|s=1}}
* {{homophones|en|base}}

====Adjective====
{{en-adj|basser}}

# Of sound, a voice or an instrument, [[low]] in
#: ''The giant spoke in a deep, '''bass''', rumbl
```

Figure 2: Extract of the MediaWiki source of the page *bass* in the Wiktionary dump. Elements between double curly braces (e.g. `{{en-adj|basser}}`) are "Templates", a kind of parameterised procedure (here, a call to template `en-adj` with argument `basser`).

Pronunciation [edit]

- enPR: bās, IPA(key): /beɪs/
- Audio (US) 0:01
- Rhymes: -eɪs
- Homophone: base

Adjective [edit]

bass (*comparative* **basser**, *superlative* **bassest**)

1. Of sound, a voice or an instrument, low in pitch or frequency.

*The giant spoke in a deep, **bass**, rumbling voice that shoo*

Figure 3: Extract of the page *bass*, as viewed in a browser, after expansion of the MediaWiki source into a valid HTML file.

This approach is usually used to extract simple information from Wiktionary, like a list of all English terms or their pronunciation, as this information is represented rather systematically using the template call `{{IPA|en|...}}`. A simple regular expression will extract this information easily and reliably.

However, this approach has several shortcomings. First, depending on the Wiktionary edition you extract from, there may be many ways to encode lexical data, as the entry structure has evolved and older entries are using older encoding conventions. In many cases, convenient templates are used to allow for a condense representation of data, but defective entries will use a specific encoding not captured by these templates. Also, the structure and encoding of Wiktionary entries evolves continually as the community updates the templates to ease entry additions. Due to this, many experiments are not reproducible as time goes by as the extraction programs become obsolete due to sometimes major changes in the Wiktionary structure.

Second, much of the information that is present in the Wiktionary HTML page is not visible in the MediaWiki source. For instance, in the excerpt of the Wiktionary *bass* page, one can find

bass (*comparative* **basser**, *superlative* **bassest**)

but this snippet is the result of the template call `{{en-adj|basser}}` where the string *bassest* does not appear. In the English Wiktionary edition, the `en-adj` template calls a Lua program[21] which computes this word form. Hence, as noted in (Ylonen, 2022), a full implementation of the Lua language (and the Scribunto[22] standard library) is required if one wants to extract most Wiktionary data[23].

This is the first approach we have attempted, and it seemed to be the most straightforward, but turned out to be inefficient: after downloading the latest Wiktionary XML dump, we wanted to extract all entries that belong to the Wiktionary category *English multiword terms*. But the category information only appears in five (badly encoded) English entries' MediaWiki source. In all other MWE entries, the categorisation is a side effect of the call of some templates appearing in the MediaWiki source. Moreover, the `https://en.wiktionary.org/wiki/Category:English_multiword_terms` page itself does not appear in the dump, as it is a special page that is computed on demand by the Wiktionary server.

Hence, in a second attempt, we tried to use the Wiktionary API to query for these categories.

3.2 Using Wiktionary API

The Wiktionary API is a RESTful interface that allows programmers to access the data contained in

[21] Such programs are called *modules* in MediaWiki. They are special pages that contain program(s) in *Lua*, a Turing complete programming language.

[22] Scribunto is the MediaWiki extension which allows for the use of any Lua program in a Wikimedia page.

[23] This was less of a problem when the language editions were not heavily depending on such modules and many of the experiments cited before will not be reproducible without this nowadays.

the Wiktionary dictionary through standard HTTP requests. It may be used to query for definitions, translations, links or categories of a specific Wiktionary page. In our cases, we planned to use it to query each page for its categories.

This would be simple if the size of Wiktionary dump was not so massive: more than 8.5 million entries need to be checked, which means 8.5 million requests sent to Wiktionary API. This is quite slow and if not done correctly will lead to being blacklisted from the Wiktionary website.

Using this approach, described in (Bajčetić et al., 2023) we have extracted over 98% of MWEs from Wiktionary and compiled a list of 153,525 MWEs without IPA, and a gold standard of 4,979 MWEs with IPA - we can see that only about 3% of MWEs have pronunciation information in Wiktionary.

However, this approach was very time-consuming and can only be applied on a specific dump. Hence, as the Wiktionary data is always growing, new MWEs introduced in Wiktionary will not benefit from this work. This is the reason why we tried to reproduce our experiment using the DBnary dataset.

3.3 Querying DBnary

DBnary (Sérasset and Tchechmedjiev, 2014; Sérasset, 2015)[24] is a lexical resource extracted from 23 language editions of Wiktionary. This dataset is structured in RDF using the *OntoLex-Lemon* model (McCrae et al., 2017), which was developed and which is further extended in the context of the W3C Community Group "Ontology Lexica".[25] The DBnary extraction program is open-source[26] and one can create issues when errors are spotted or additional information is required.

With DBnary, the whole set of lexical information extracted from the 23 language editions of Wiktionary may be seen as a huge graph that can be downloaded and queried online using the SPARQL language[27] or accessed interactively through a faceted browser.[28] Moreover, any node (Page, Lexical Entry, Lexical Sense, Translation, Word Form, etc.) in this huge graph is designed by a unique URI[29] that may be dereferenced (i.e. accessed through the HTTP protocol) so that any person or process can obtain its related information easily which is compliant to the guidelines of the Linguistic Linked Open Data (LLOD) framework (Declerck et al., 2020).[30] Using DBnary is a matter of crafting SPARQL queries and evaluating them using a public endpoint.

By our first use of DBnary, we saw that, while pronunciation information is available, some of the information we required was missing from the English dataset:

- the entries were only typed as `ontolex:LexicalEntry` and no finer grain typing (as `ontolex:Word`, `ontolex:MultiWordExpression` or `ontolex:Affix`) was available,
- derivation information between terms was not extracted.

These missing elements were added and are now available in versions starting from February 2023. The extraction program now correctly *types* English Wiktionary entries either as `ontolex:Word` or as `ontolex:MultiWordExpression`. Moreover, derivation relations are now extracted and available in the graph using `dbnary:derivesFrom` transitive property.

Figure 4 shows an example of the organisation of two heteronym lexical entries described by the same page, along with their canonical forms (with written and phonetic representation).

Figure 4 also shows how the derivation relation is modelled in DBnary, using the transitive `dbnary:derivesFrom` property. It must be noted that in Wiktionary original data, the derivation links point to Wiktionary pages but not to Wiktionary entries, hence, the DBnary modelling reflects this as it is usually difficult to automatically

[24] See `http://kaiko.getalp.org/about-dbnary/` for the current state of development of DBnary.

[25] See `https://www.w3.org/community/ontolex/` for more details.

[26] `https://gitlab.com/gilles.serasset/dbnary`

[27] SPARQL is the "standard query language and protocol for Linked Open Data on the web or for RDF triplestores", quoted from `https://www.ontotext.com/knowledgehub/fundamentals/what-is-sparql/`. The SPARQL endpoint of DBnary can be accessed at `http://kaiko.getalp.org/sparql`

[28] The browser can be accessed at `http://kaiko.getalp.org/fct/`

[29] E.g. the URI `http://kaiko.getalp.org/dbnary/eng/bass` represents the Wiktionary *Page* bass that further *describes* different *Lexical Entries* (In English, one adjectival, one verbal and three nominal and eleven others in nine other languages.)

[30] See also `http://www.linguistic-lod.org/`.

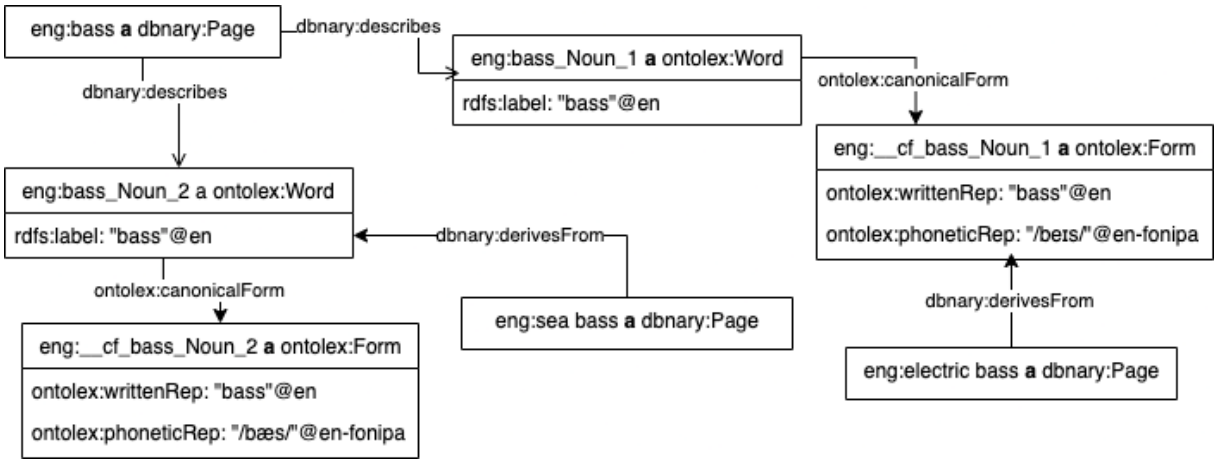

Figure 4: A very small extract of the DBnary graph showing DBnary page *bass* and two of the lexical entries it describes (bass_Noun_1 [sound, music, instrument] and bass_Noun_2 [perch, fish]) and their respective canonical forms. The pages *sea bass* and *electric bass* are also represented with their derivation relations.

choose which lexical entry(ies) is (are) the valid target of the derivation relation. But, applying the property in the inverse direction (could be named `dbnary:derivesTo`), the subject/source of the relation is a lexical entry within a Wiktionary page, pointing to a MWE page. As MWE pages consist mainly of only one lexical entry, we can precisely establish a "subterm" relation between a single lexical entry and the MWEs it occurs in, combining if needed both "directions" of use of the property. This point is very important, as it allows projecting all the lexical information of the single lexical entry to the component it builds within a MWE, as this is briefly presented in Section 5.

In the DBnary representation of Wiktionary we find lexical entries (including words, MWEs or affixes), their pronunciation (if available in Wiktionary), their sense(s) (definitions in Wiktionary), example sentences and DBnary glosses, which are offering a kind of "topic" for the (disambiguated) entries, but those glosses are not originated in the category system of Wiktionary. The glosses are taken from available information used to denote the lexical sense of the source of the translation of an entry from English to other languages.

DBnary does not extract Wiktionary categories, as most of these are implicit in the MediaWiki code and are the result of the full processing of the MediaWiki source. This processing is too heavy to compute for the 8.5M+ pages found in the English Wiktionary edition. Without this full processing, the extraction process takes almost 14 hours on a recent CPU server, more than 70% of which goes in the execution of Lua Modules. As this extraction has to be re-computed twice a month as new dumps are released, taking several days for such an extraction is not worth it.

In the paper, we reproduce the approach described in (Bajčetić et al., 2023), using only DBnary data. The added value of using DBnary comes from the fact that the data is updated twice a month and extractors are usually maintained to reflect changes in Wiktionary representation of the entries. Hence, reproducing this work will be possible without a high data preparation cost, and future MWEs described in future versions of Wiktionary will benefit of it.

4 Enriching pronunciation for MWEs using DBnary

4.1 Assessing the size of the problem

Before proceeding to the experiment using DBnary data[31], we first probe the dataset to see if it faithfully reflects the Wiktionary data. First, we would like to know how many entries have a canonical form with pronunciation, using the SPARQL query displayed in Listing 1.[32]

```
SELECT ?mweOrLE, COUNT(?e)
FROM <http://kaiko.getalp.org/dbnary/eng>
WHERE {
  ?e a ?mweOrLE ;
     ontolex:canonicalForm ?wf.
    FILTER
      exists {?wf ontolex:phoneticRep ?pr}.
```

[31]These figures and the whole experiment is available in a notebook at https://github.com/serasset/dbnary-mwt-pronunciations/blob/main/notebooks/MWE_Pronunciation_LDK2023.ipynb.

[32]Note that in all SPARQL queries, we do not add the PREFIXes as they are known and optional on the DBnary server.

```
    VALUES ?mweOrLE
           { ontolex:MultiWordExpression
             ontolex:LexicalEntry}
} GROUP BY ?mweOrLE
```

Listing 1: SPARQL query to count the available phonetic representations (`?pr`) of lexical entries (`?e`). We also get the counts for entries types as `ontolex:MultiWordExpression` or `ontolex:LexicalEntry`.

A similar query is used to count the entries without pronunciation information. The results are given in Table 1.

type	**with (# of pron)**	**without**
LE	107327 (173512)	1102485
MWE	4977 (8143)	214243

Table 1: The number of English Lexical Entries available in the English Wiktionary with or without pronunciation information, among which we also count the MWEs. The total number of distinct pronunciations is also given.

These values are slightly different from the ones obtained using the Wiktionary category pages or the statistics pages. The reasons for this are (1) the Wiktionary statistics have been done a year ago, while the DBnary query reflects the status of the latest dump[33] and (2) Wiktionary categories refer to *pages* while the figures we have here are referring to *lexical entries* (there are usually several lexical entries described in a single page[34]).

Despite being marginally different, these counts confirm the original observed proportions of less than 10% of Lexical Entries having pronunciation, while less than 2.3% of MWEs come with pronunciation information.

4.2 Borrowing pronunciation of MWEs from their components

The main idea in (Bajčetić et al., 2023) is to construct the pronunciation of MWEs by borrowing the pronunciation of their components. This is straightforward when components have a single pronunciation, but requires care when the pronunciation differs for different meanings (in the case of heteronyms).

To compute its pronunciation, the MWE is decomposed in components and each component is independently queried for its pronunciation information. For this experiment, the decomposition has been done straightforwardly by breaking the MWE according to spaces and assuming that each component of the derivation is a canonical form.

As components may have several pronunciations, all the resulting pronunciations are combined leading to a set of candidates. However, this method is faulty when we are dealing with heteronyms.

4.3 Dealing with heteronymy

As defined on Wikipedia, "*a heteronym (also known as a heterophone) is a word that has a different pronunciation and meaning from another word but the same spelling*".[35] A common example for heteronyms is given by the lexical entries "bass" (fish, pronounced /bæs/) and "bass" (sound, low in pitch, pronounced /beɪs/).

In our setup, heteronyms are defined as *pages describing* at least two *lexical entries* which have at least two different sets of pronunciations. To identify those heteronyms, we query all pages for their different pronunciation sets using the SPARQL query given in Listing 2. In the resulting table, the heteronyms are pages that appear more than once.

```
SELECT ?p ?prons
       (GROUP_CONCAT(?e; SEPARATOR = ",")
          as ?entries)
FROM <http://kaiko.getalp.org/dbnary/eng>
WHERE {
  ?p a dbnary:Page; dbnary:describes ?e.
  {
    SELECT ?e              ## sub query 1
       (GROUP_CONCAT(?pr ; SEPARATOR=",")
          as ?prons) {
      SELECT ?pr ?e {      ## sub query 2
        ?e ontolex:canonicalForm /
           ontolex:phoneticRep ?pr .
      } GROUP BY ?e ?pr
        ORDER BY ?pr
    } GROUP BY ?e
  }
```

[33] These numbers reflect the DBnary dataset version 20230320. As Wiktionary evolves and DBnary dataset is updated, more data is constantly added to the resource. For instance, the previous version (dated 20230301), contained 172846 (resp. 1097873) Lexical entries with (resp. without) pronunciation and 8074 (resp. 213276) MWEs with (resp. without) pronunciation.

[34] For instance, the 173512 lexical entries with pronunciation counted here are described in 75082 different pages.

[35] Quoted from https://en.wikipedia.org/wiki/Heteronym_(linguistics) [accessed 2023.03.37]

```
} GROUP BY ?p ?prons
```

Listing 2: SPARQL query to extract all heteronym pages (?p), along with their distinct pronunciations (?prons) and the corresponding entries (?entries). Sub-query 1 and 2 extract and group the different pronunciations for each lexical entry, then entries are grouped by distinct pronunciation set.

Page	Pronunciations	gloss
911	/ˌnaɪn wʌn ˈwʌn/	emergency
911	/ˈnaɪn əˌlɛvən/	porsche
bass	/beɪs/	low pitch
bass	/bæs/	fish
hinder	/ˈhaɪn.də/,/ˈhaɪn.dɚ/	make difficult
hinder	/ˈhɪndə/,/ˈhɪndɚ/	more hind
tower	/ˈtaʊ.ə(ɹ)/,/ˈtaʊɚ/	tall structure
tower	/ˈtəʊ.ə(ɹ)/	one who tows
lead	/lid/, /liːd/	to guide
lead	/lɛd/	metal

Table 2: A sample of heteronym pages along with their dictinct pronunciation groups.

In English DBnary, we identified 970 heteronym pages among the 75082 pages with pronunciation. A sample of these is given in table 2.

When a component is identified as a heteronym, we have to choose among the different pronunciations for the one that is valid for the MWE. For example, in the MWE *lead pencil*, the component *lead* corresponds to the metallic sense, pronounced /lɛd/, while in *lead astray*, the component *lead* corresponds to the verbal "to guide" sense, pronounced /liːd/. The same phenomenon occurs for *bass guitar* where *bass* refers to the "low in pitch" meaning, pronounced /beɪs/, while sea bass contains the *bass* (as a fish) component, pronounced /bæs/.

In order to correctly decide which pronunciation should be used for such a heteronym component and not over-generate erroneous pronunciations, we use the derivation relations that are present in Wiktionary and are now available in DBnary. Figure 4 shows an example of such derivation relation in the context of the heteronym page *bass*. All derivation relations is extracted from DBnary with the SPARQL query given in Listing 3. The English DBnary dataset contains 239284 such relations.

```
SELECT
  DISTINCT ?deriv_from ?source_label
           ?deriv_to ?target_label
FROM <http://kaiko.getalp.org/dbnary/eng>
WHERE {
  ?deriv_to
    dbnary:derivedFrom ?deriv_from ;
      dbnary:describes
          / rdfs:label ?target_label .
  ?deriv_from rdfs:label ?source_label.
}
```

Listing 3: SPARQL query to extract all derivation relations from DBnary

When a component of a MWE is a heteronym, we look for a corresponding derivation relation that points us to the *Lexical Entry* the MWE derives from. We then use the pronunciation of this Lexical Entry and ignore pronunciations of other Lexical Entries with the same canonical form.

4.4 Experiment and evaluation

In order to evaluate this experiment, we will use the pronunciations of the 4977 MWEs that are available in DBnary as a gold standard. When computing the pronunciation candidates, four cases are used:

- **NP**: No pronunciation is available for at least one of the components,
- **COMP**: All components are non-heteronym and have pronunciation information,
- **HCOMP**: At least one component is a heteronym and derivation relation is available,
- **HND**: At least one element is heteronym and no derivation relation is available.

In **NP** and **HND** cases, we chose not to produce any candidates. We measure the Precision, recall and F1-measure in cases **COMP** and **HCOMP** by comparing known pronunciation with produced candidates. For this comparison, we applied four normalisation methods on the pronunciations:

- **NO**: pronunciation strings are compared without any normalisation,
- **SPA**: spaces are removed from pronunciation strings before comparison,
- **SUP**: suprasegmental signs (primary and secondary stresses, lengths, syllable breaks, etc.) are removed from the pronunciation strings before comparison,
- **SUPSPA**: suprasegmentals and spaces are removed from the pronunciation strings before comparison.

	COMP			HCOMP			All[a]		
Norm	*prec*	*recall*	*f1*	*prec*	*recall*	*f1*	*prec*	*recall*	*f1*
NO	.1172	.1731	.1269	.0310	.0781	.0381	.0516	.0771	.0560
SPA	.1186	.1761	.1285	.0382	.0976	.0481	.0524	.0789	.0570
SUP	.2937	.5045	.3324	.1688	.3993	.2057	.1318	.2292	.1495
SUPSPA	.3457	.5994	.3896	.2367	.5712	.2938	.1561	.2748	.1766

[a]Overall performance accounting for cases where we do produce results (COMP and HCOMP) and cases where we do not (NP, HND). This is given for exhaustive evaluation, but as we were able to distinguish between the different cases, these measure do not reflect the real difficulty of the task.

Table 3: Evaluation of the experiments using four normalisations on the pronunciation strings.

case	**in gold standard**	**in DBnary**
NP	2448	86689
COMP	2160	114969
HCOMP	128	2246
HND	241	10340

Table 4: The number of MWE in each of the different evaluation cases.

Table 3 gives the precision, recall and F1-measure for the different cases and normalisations. We give overall evaluation results on all four cases for exhaustivity, but as the process is generating pronunciation proposals that will be manually validated, the figures only reflect the proportion of cases where we can propose something (54.7%) and cases where we cannot (45.3%). Overall, this evaluation shows encouraging results when ignoring the suprasegmental elements of the pronunciation strings, thus validating the main strategy to raise the number of pronunciations for MWEs by borrowing pronunciations from their components. However, suprasegmental seems harder to figure out and we hypothesise that they are as much influenced by the global MWE context than by each intra-component pronunciation.

As detailed in table 4, overall, we are able to produce pronunciation candidates for 114969 MWEs using the **COMP** strategy and for 2246 MWEs using the **HCOMP** strategy.

4.5 Lessons learned and current work

By using DBnary dataset we were able to more easily extract lexical data on which we applied the original strategy described in (Bajčetić et al., 2023). This process is quite efficient and does not require any manual intervention and may be used each time new MWEs are added to Wiktionary.

However, we currently identify several shortcomings for which we should investigate deeper. The first limitation we need to address is identifying to which extent the proposed strategy may be ported to other languages available in DBnary (which currently extract from 23 different editions). In this experiment decomposition of the MWE in a set of component is simply based on space characters and we assumed that each component appeared in its canonical form. Such heuristics seem justified in the case of English language where entries have very few inflected forms, but will certainly become questionable if we apply it on other languages like French (that has a more productive morphology) or German (where components are usually concatenated without spaces). Moreover even in the case of English language, with this heuristic the term *acoustic bass guitar* cannot be decomposed as "*acoustic*" + "*bass guitar*" and we cannot take advantage of the already existing pronunciation attached to "*bass guitar*". Future work should investigate other decomposition processes and the use of inflected forms as components in a second step.

Another limitation, that may explain the precision measures, comes from the fact that DBnary does not correctly identify the regional variant information of pronunciation strings. For example, when computing pronunciation for *bomb crater* we look for the entries *crater* (UK: /kɹeɪ.tə(ɹ)/ , US: /kɹeɪ.tɚ/) and *bomb* (UK: /bɒm/, US: /bɑm/, obsolete: /bʌm/) and produce six candidates that are the combination of all individual components pronunciation, while only two should be produced by combining the UK (resp. US) pronunciations. This shortcoming will not be addressed before DBnary corrects its English extractor to properly identify and represent the regional variant for each extracted pronunciation.

5 Semantic enrichment of components of MWEs

The former sections demonstrated the advantage of concentrating our work on adding pronunciation information to MWEs on the use and adaptation of the DBnary resource. We stressed that DBnary is offering the extracted information from Wiktionary in a structured fashion, more precisely using LOD compliant models and vocabularies. And we see in this feature another precious advantage of using DBnary for our work dealing with the enrichment of MWEs included in Wiktionary (and in the longer term also for resources like the Open English WordNet, or others), focusing in a next step on morphosyntactic and semantic information that can be added to the components of such MWEs.

5.1 The decomposition module of OntoLex-Lemon

As DBnary is making use of the *OntoLex-Lemon* model, we can take advantage of the existence of its "Decomposition" module,[36] which is graphically displayed in Figure 5.

We can observe that the property "decomp:subterm" of the Decomposition module is equivalent to the property "dbnary:derivesFrom", recently introduced in DBnary, in order to represent the Wiktionary section "Derived terms" (see Figure 4) for comparison. Therefore, we can just map the "rdf:Object" of "dbnary:derivesFrom" to the "rdf:Object" of "decomp:subterm", while the rdf:Subject of "decomp:subterm" is the MWE itself, as been seen in Listing 4.

As a result, the recent adaptations of DBnary allow not only to generate pronunciation information for MWEs contained in the English edition of Wiktionary, but also to add morphosyntactic and semantic information to the components of such MWEs, and to encode this information in such a way that the new data set can be published on the Linguistic Linked Open Data cloud.

```
:electric_bass_lex a
      ontolex:MultiwordExpression ;
  decomp:subterm eng:electric_Adjective_1 ;
  decomp:subterm :eng:bass_Noun_1 .
```

Listing 4: The (simplified) representation of "electric bass" using the Decomposition module of *OntoLex-Lemon*, with links to lexical data encoded in DBnary

Using this module, we can thus explicitly encode the morphosyntactic, semantic and domain information of the components of MWEs, which are only implicitly present in Wiktionary. For our example, we know that "electric" has PoS "adjective" (Wiktionary lists also a nominal use of the word) and "bass" the PoS "noun" (Wiktionary lists also an adjectival and a verbal uses), while semantically disambiguating the components of the MWE (in the full DBnary representation, the "ontolex:Word": "eng:bass_Noun_1" is linked to the corresponding instances of "ontolex:Sense". And in fact, we can then link to a corresponding Wikidata entry for "bass guitar" (https://www.wikidata.org/wiki/Q46185) and the one for "electricity" (https://www.wikidata.org/wiki/Q12725)

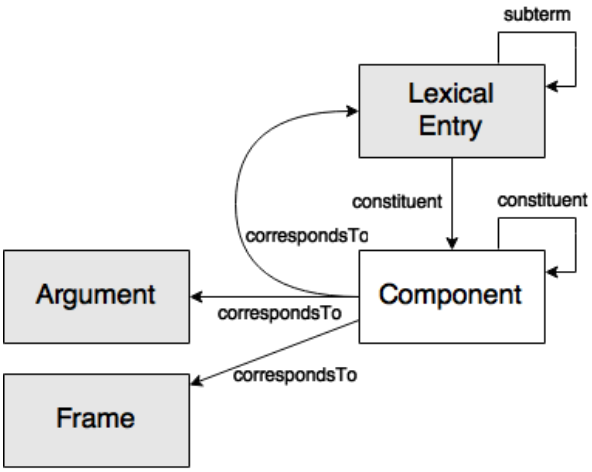

Figure 5: The Decomposition module of *OntoLex-Lemon*. Taken from https://www.w3.org/2016/05/ontolex/#decomposition-decomp

6 Conclusion and future work

We described in this paper on-going work on computing pronunciation information for multiword expressions (MWEs) included in Wiktionary. In the course of this work, we got acquainted with the DBnary resource, which is offering a Linked Open Data compliant representation of lexical information extracted from Wiktionary, using at its core the *OntoLex-Lemon* model and other related

[36] The specification of *OntoLex-Lemon* describes "Decomposition" in those terms: "Decomposition is the process of indicating which elements constitute a multiword or compound lexical entry. The simplest way to do this is by means of the subterm property, which indicates that a lexical entry is a part of another entry. This property allows us to specify which lexical entries a certain compound lexical entry is composed of.". Taken from https://www.w3.org/2016/05/ontolex/#decomposition-decomp

vocabularies. As it was immediately clear that using the extraction engine of DBnary is easing massively our work, we teamed with the maintainer of DBnary, who adapted the extraction engine for our needs. Those recent updates are the focus of this paper. We discovered also that this way, we can not only easily generate pronunciation information for MWEs, but we can also in a straightforward manner add morphosyntactic and semantic information to the components of MWEs. This will lead to the generation of a new data set for English MWEs. As a result, the DBnary engine is now more than an extractor from Wiktionary and a mapper to an LOD compliant representation, as it generates lexical information that can be used for enriching existing lexical resources.

We plan to port some of our approach to other languages supported by DBnary, aiming at a multilingual data set for MWEs.

Limitations

While our approach can probably be transferred to other languages, in cases where the Wiktionary structure for those languages is similar, there is one aspect of pronunciation extraction and combination that we have not discussed and this concerns the pronunciation(s) of variants of English, which are included in Wiktionary, like British, General American, Irish, Canadian, Australian and New Zealand English. In our current work we ignored the variants as they were not (yet) available in DBnary, so we "overlook" the variants information and produce potentially unusable new pronunciations (that will have to be discarded at manual validation). However, we would want to include all these varieties of our future work. This should not be too complicated, as the approach would follow the same principle as explained in the paper, with one extra layer of variant matching.

Another limitation of our work lied in the fact that Wiktionary is ever-changing. So anything done at one point in time needs to be re-done in the future due to changes in the data and also newly added data. The fact that Wiktionary grows quite fast means that the best approach would be incremental or recursive in some way, and automatically check for newly added pronunciations which can create new MWEs pronunciations, while also confirming that the previously created ones have not been altered and need updating. But our teaming with the maintainer of DBnary seems to offer a good solution, as DBnary is updated twice a month.

Another current limitation lies in the fact that we consider only binary MWEs. This is due in a good part to the fact that Wiktionary is not delivering a lot of information when dealing with longer MWEs, but we are analysing the available data in more details.

Ethics statement

We consider our work to have a broad impact because Wiktionary is widely used across the world, as a free and open-source resource. Additionally, we plan to include the output of our research into other resources, like for example the Open English WordNet, which are also resources that are free to use and open-source. We hope that in this way the results of our work can potentially be useful to people all around the world who read or speak English, as well as text-to-speech (and possibly speech-to-text) systems which are gaining popularity and are very important for the visually impaired community, among others.

We do not see any ethical issue related to the generation of additional information that can be attached to Wiktionary MWEs and their components.

Acknowledgements

The presented work is pursued in the context of the COST Action NexusLinguarum – European network for Web-centered linguistic data science (CA18209), 731015). The DFKI contribution is also pursued in the context of the LT-BRIDGE project, which has received funding from the European Unions Horizon 2020 Research and Innovation Programme under Grant Agreement No 952194.

References

Lenka Bajčetić, Thierry Declerck, and Gilles Sérasset. 2023. Enriching multiword terms in Wiktionary with pronunciation information. In *Proceedings of the 19th Workshop on Multiword Expressions (MWE 2023)*, pages 65–72, Dubrovnik, Croatia. Association for Computational Linguistics.

Christian Chiarcos and Maria Sukhareva. 2015. OLiA Ontologies of Linguistic Annotation. *Semantic Web*, 6(4):379–386. Publisher: IOS Press.

P. Cimiano, P. Buitelaar, J. McCrae, and M. Sintek. 2011. Lexinfo: A declarative model for the lexicon-ontology interface. *Journal of Web Semantics*, 9(1):29–51.

Philipp Cimiano, John McCrae, and Paul Buitelaar. 2016. Lexicon Model for Ontologies: Community Report, 10 May 2016. Technical report, W3C.

Gerard de Melo. 2015. Lexvo.org: Language-related information for the Linguistic Linked Data cloud. *Semantic Web*, 6(4):393–400.

Thierry Declerck and Lenka Bajčetić. 2021. Towards the addition of pronunciation information to lexical semantic resources. In *Proceedings of the 11th Global Wordnet Conference*, pages 284–291, University of South Africa (UNISA). Global Wordnet Association.

Thierry Declerck, John Philip McCrae, Matthias Hartung, Jorge Gracia, Christian Chiarcos, Elena Montiel-Ponsoda, Philipp Cimiano, Artem Revenko, Roser Saurí, Deirdre Lee, Stefania Racioppa, Jamal Abdul Nasir, Matthias Orlikowsk, Marta Lanau-Coronas, Christian Fäth, Mariano Rico, Mohammad Fazleh Elahi, Maria Khvalchik, Meritxell Gonzalez, and Katharine Cooney. 2020. Recent developments for the linguistic linked open data infrastructure. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5660–5667, Marseille, France. European Language Resources Association.

John P. McCrae, Paul Buitelaar, and Philipp Cimiano. 2017. The OntoLex-Lemon Model: development and applications. In *Proc. of the 5th Biennial Conference on Electronic Lexicography (eLex)*.

John Philip McCrae, Alexandre Rademaker, Ewa Rudnicka, and Francis Bond. 2020. English WordNet 2020: Improving and extending a WordNet for English using an open-source methodology. In *Proceedings of the LREC 2020 Workshop on Multimodal Wordnets (MMW2020)*, pages 14–19, Marseille, France. The European Language Resources Association (ELRA).

Gilles Sérasset. 2015. Dbnary: Wiktionary as a lemon-based multilingual lexical resource in rdf. *Semantic Web*, 6:355–361.

Gilles Sérasset and Andon Tchechmedjiev. 2014. Dbnary : Wiktionary as Linked Data for 12 Language Editions with Enhanced Translation Relations. In *3rd Workshop on Linked Data in Linguistics: Multilingual Knowledge Resources and Natural Language Processing*, page to appear, Reyjkjavik, France.

Tatu Ylonen. 2022. Wiktextract: Wiktionary as machine-readable structured data. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1317–1325, Marseille, France. European Language Resources Association.