DMLEX on Wikibase: Legacy dictionaries as collaboratively editable dataset

Simon Krek,[1,2] Primož Ponikvar,[3] Andraž Repar,[2] Iztok Kosem,[1,2] David Lindemann[4]

[1]University of Ljubljana (Faculty of Arts & Faculty of Computer and Information Science), Ljubljana, Slovenia
[2] Jožef Stefan Institute, Ljubljana, Slovenia
[3] Centre for Language Resources and Technologies at the University of Ljubljana, Ljubljana, Slovenia
[4] EHU University of the Basque Country, Vitoria-Gasteiz, Spain
E-mail: simon.krek@ijs.si, pponikvar@yahoo.com, andraz.repar@ijs.si, iztok.kosem@ijs.si, david.lindemann@ehu.eus

Abstract

This paper presents an experimental workflow for converting legacy digitized dictionaries into the DMLex standard and subsequently importing them into a Wikibase instance. DMLex, a serialization-independent model developed by the OASIS LEXIDMA Technical Committee, aims to provide a universal and modular representation of lexicographic data. The study tested whether dictionaries from heterogeneous sources—originally encoded in internal XML formats—could be reliably transformed into DMLex-compliant representations and repurposed for collaborative editing and enrichment on a structured linked data platform. The transformation was achieved through a combination of rule-based scripts, manual refinement, and large language model assistance. While DMLex proved adaptable to a wide range of lexical phenomena, several limitations became apparent during the Wikibase integration phase. These findings suggest that practical deployment of DMLex benefits from clearer conventions and validation strategies when applied beyond theoretical modeling. The results confirm DMLex's potential for future-proof dictionary modeling, while also highlighting areas where further specification and community consensus are needed to support its application in digital infrastructures and collaborative environments.

Keywords: legacy dictionaries; conversion; standardization; semantic web; linked data

1. Introduction

Over the last decade lexicography has moved decisively from printed pages to interconnected digital ecosystems. Large corpora, language-technology pipelines and—most recently—large language models (LLMs) are reshaping the entire lexicographic workflow. This paper presents a pipeline for converting legacy dictionaries into the DMLex intermediate format and publishing them in a Wikibase instance for collaborative curation.

The legacy dictionaries addressed in this study were acquired within the national *RSDO*[1] infrastructure project and subsequently released as open-source data under the *CC BY-SA 4.0* license. Although already digitised, they still existed in a legacy workflow: the material had been compiled for print, laid out with desktop-publishing software, and only later

[1] rsdo.slovenscina.eu

exported to a bespoke, project-specific XML schema. To integrate these resources into modern, tool-agnostic environments we first converted the custom XML to the emerging OASIS Open standard DMLex, and then modelled DMLex elements in a dedicated Wikibase so that lexicographers and the wider community can edit, extend and link the entries collaboratively.

The remainder of the paper is organized as follows. Section 2 reviews related work on legacy dictionary conversion and lexical data standards. Section 3 introduces the datasets used in the experiment. Section 4 outlines the two-step processing pipeline: the first step converts the heterogeneous source formats into DMLex-compliant XML, while the second step maps DMLex elements onto Wikibase entities and statements. Section 5 provides a summary of the overall process and discusses key challenges encountered, as well as directions for future development.

2. Related work

Several recent projects have tackled the challenge of converting legacy dictionaries into structured, standardized digital formats, often using TEI or linked data models. The Digital Historical Dictionary of Italian project (Biffi et al., 2019) demonstrates how iterative pattern-based extraction techniques can be used to convert scanned dictionaries into structured TEI XML, maintaining both typographic features and semantic structure. The Academia das Ciências de Lisboa Dictionary case study (Salgado, 2018) describes a semi-automated conversion from PDF to TEI Lex-0, relying on structured templates and normalization pipelines to ensure consistency across entries. In the MORDigital project (Almeida et al., 2022), a historical Portuguese dictionary was encoded in TEI Lex-0 and linked to external semantic resources via OntoLex-Lemon, illustrating the enrichment potential of linked data integration. The Abaev project (Belyaev et al., 2021) provides a detailed account of converting a complex multilingual print dictionary into TEI, highlighting the importance of distinguishing metalanguage and object language in digital encodings. Maxwell and Bills (Maxwell & Bills, 2017) focus on the digitization of endangered language dictionaries using OCR and structured markup, emphasizing the need to adapt encoding strategies to the linguistic and typographic diversity of legacy sources. In addition, initiatives such as Grobid (Romary & Lopez, 2015) and Elexifier (Tiberius et al., 2021) provide environments for semi-automated, computer-assisted conversion of dictionary data into structured digital formats.

There have been several attempts at standardizing digital dictionaries. TEI-Lex0 (Tasovac et al., 2018) is a simplified and consistent version of the TEI guidelines made specifically for dictionaries, so that different projects can use the same rules and tools. OntoLex-Lemon (McCrae et al., 2017) is a standard for publishing dictionary data as linked data on the web, using the RDF format, which makes it easy to connect words and meanings across different datasets. Lexical Markup Framework (Francopoulo, 2013) is an ISO standard that gives a general model for building digital lexicons and supports many kinds of language data.

3. Data

The project builds on six legacy bilingual dictionaries that were acquired under the national RSDO infrastructure programme and released as open data under a permissive cc-by-sa 4.0

licence. All of them were originally compiled for print and later converted into a bespoke XML format.

Slovensko-srbskohrvaški slovar (Slovenian → Serbo-Croatian). 2004. author J. Jurančič.
Srbskohrvatsko-slovenski slovar (Serbo-Croatian → Slovenian). 2005. author J. Jurančič.
Slovensko-angleški slovar (Slovenian → English). 1999. authors A. Grad, H. Leeming.
Priročni slovensko-angleški slovar (Slovenian → English). 2010. authors K. Grabnar *et al.* Only the SL-EN part was processed.
Veliki angleško-slovenski slovar (English → Slovenian). 1978. authors A. Grad *et al.*
Veliki slovensko-nemški slovar (Slovenian → German). 1999. authors D. Debenjak *et al.*

Snippet 12.1 shows a fragment of the bespoke XML structure that all dictionaries share.

```
<X>*********************** +V -- 744722</X>
<GS fq="3">
  <IZ>
    <I>a</I>
    <IS></IS>
  </IZ>
  <ZG>
    <BV>vez.</BV>
  </ZG>
  <PR>
    <PU>
      <P>but, however</P>
    </PU>
  </PR>
</GS>

<X>*********************** +S -- 3827</X>
<GS fq="1">
  <IZ>
    <I>abeceda</I>
    <IS>abec{e/}da</IS>
  </IZ>
  <ZG>
    <BV>sam.</BV>
  </ZG>
</GS>
```

Listing 12.1: Condensed fragment of the legacy XML structure

4. Processing

In this section we describe the two processing steps that take our six legacy resources from bespoke XML to a collaboratively editable knowledge graph. Instead of going directly from the source XML to Wikibase, we decide to use a pivot in the form of DMLex, a new lexicographic standard which introduces *a modular, IT-friendly, and content-rich data model designed to meet the needs of both lexicographers and technology developers.* DMLex *has been designed to be easily and straightforwardly implementable in XML, JSON, RDF,* *NVH, as a relational database, and as a Semantic Web triplestore.*[2] In addition to being released under an open license, the dictionaries will be provided in a standardized, digitally accessible format to ensure easy integration and use by lexicographers and other end users.

4.1 Legacy XML to DMLex transformation

The migration from legacy XML into DMLex went along three parallel tracks:

- using custom scripts to treat tags that had clear one-to-one mapping to DMLex
- skilled lexicographers carrying out quick low-effort edits
- using a large language model whenever the required human work would grow too heavy

While a full account of the challenges encountered is beyond the scope of this paper — given that each dictionary had its own idiosyncrasies and special cases — we provide below some representative examples of common transformations. The outcome of this mixed strategy is a set of schema-valid DMLex XML files ready for import into Wikibase without revisiting the legacy quirks.

1. Straightforward element mappings. A lookup table was prepared that defined one-to-one correspondences between old tags and their DMLex equivalents. For instance, the legacy grammar label `<BV>sam.</BV>` is rewritten as a `<partOfSpeech>` node, and the headword container `<G>` becomes a `<headword>` element in DMLex. An example is shown in snippet 12.2.

```xml
<!-- legacy -->
<BV>sam.</BV>
<G>anglikanski</G>

<!-- DMLex -->
<entry>
  <headword>anglikanski</headword>
  <gramGrp>
    <partOfSpeech tag="noun" listingOrder="1"/>
  </gramGrp>
</entry>
```

Listing 12.2: Straightforward mapping

2. List splitting. Legacy `<TRO>` tags often pack several translations into a single comma-separated string which must be split into individual DMLex objects. An example is provided in snippet 12.3.

```xml
<!-- legacy -->
<TRO>žkriarski, škrstaki</TRO>

<!-- DMLex -->
<sense>
  <headwordTranslation text="žkriarski"/>
  <headwordTranslation text="škrstaki"/>
</sense>
```

Listing 12.3: Comma-separated translations

[2] https://www.oasis-open.org/2025/05/29/dmlex-approved-as-oasis-standard/

3. Tilde expansion. Printed dictionaries often save space by replacing the headword within collocations with a tilde or a similar character. A significant part of the transformation process was dedicated to resolving these placeholders and restoring the full forms. An example is provided in snippet 12.4.

```
<!-- legacy -->
<G>anglik{a/}nski</G> <O>-a</O> <O>-o</O> anglikanski<FR>: <I>-a cerkev</I></FR>

<!-- DMLex -->
<sense>
  <example>
    <text>anglikanska cerkev</text>
  </example>
</sense>
```

Listing 12.4: Tilde expansion

4. Suffix expansion. Where a dash hints at predictable suffixes (e.g. -*čki*), the transformation involved combining the common stem and the ending and outputting a separate DMLex element for each form. See example in snippet 12.5.

```
<!-- legacy -->
<TRO>čjansenistian, č-ki</TRO>

<!-- \textsc{DMLex} -->
<sense>
  <headwordTranslation langCode="sh" text="čjansenistian"/>
  <headwordTranslation langCode="sh" text="čjansenistiki"/>
</sense>
```

Listing 12.5: Dash-based suffix expansion

5. Issues with complex legacy content. While most legacy tags were relatively easy to map onto DMLex, some cases proved more challenging. One such example is shown in snippet 12.6, where metadata—such as indicators, definitions, collocators, and domain, style, or regional labels—precedes individual translations. This type of information does not always align cleanly with the sense structure defined by DMLex. In DMLex, such metadata can only be represented using the definition or label elements. While this is adequate at the sense level, it does not allow for open-ended metadata to be associated directly with individual translations. As a result, we could either omit this information or restructure the entry. To preserve the full richness of the metadata, we opted to split the original sense into multiple, more granular senses. Although this approach diverges from the structure of the source dictionary, it ensures that all metadata can be explicitly represented. In doing so, we prioritized completeness and fidelity of information over strict adherence to the original macrostructure.

```
<GN>agregat</GN>
 <S>
<TR1/><KP>tehn.</KP> <IN>za proizvajanje čmoi</IN> <TR>engine</TR>
<TR1/><KP2>tehn.</KP2> <KO>pogonski, hladilni</KO> <TR>unit</TR>
<TR1/><KP2>tehn.</KP2> <IN>za proizvajanje elektrike</IN> <KO>čelektrini</KO> <TR>generator</TR>
</S>
 <S>
<TR1/><KP>ekon.</KP> <TR>aggregate</TR>
<FR><FRB>denarni agregat</FRB> <FRP>monetary aggregate</FRP></FR>
```

```
</S>
```

Listing 12.6: Metadata on the translation level in legacy dictionaries

4.1.1 LLM-augmented workflows

Whenever the manual workload became too heavy, we delegated the task to an LLM (specifically OpenAI GPT-4o), primed with in-context examples so that it could return the desired output in one shot. An example of a full prompt—including the system role, the detailed instructions, and the six demonstration cases—is shown in snippet 12.7.

```
[
  {"role": "system",
    "content": "You are a lexicographer specialized in Slovenian
               dictionaries. You return JSON data." },
  {"role": "user",
    "content": "You will receive three pieces of information: a dictionary headword in Slovenian,
          a Slovenian example phrase or a German example phrase. The Slovenian example phrase is
         sometimes not complete and is left to the reader to extrapolate the actual phrase based
         on the headword. In other cases, it is complete and you need to return it as is. Always
         compare the German example with the Slovenian example and make sure they are equivalent.
          You need to generate a new Slovenian example phrase that will be a proper equivalent of
          the German phrase." },
  {"role": "user",
    "content": "There are several different cases you need to consider: 1) headword: angorski,
         sl example: čmaka, de example: Angorkatze. In this case, you need to return angorska
         čmaka. 2) headword: aprobiran, sl example: aprobiran veterinar, de example: approbierter
          Tierarzt. In this case, your need to return aprobiran veterinar. 3) headword: avdienca,
          sl example: pri, de example: bei. In this case, you need to return pri. 4) headword:
         babica, sl example: pavja babica, de example: Pfauenschleimfisch. In this case, you need
          to return pavja babica. 5) headword: baje, sl example: baje je bolan, de example: er
         soll krank sein. In this case, you need to return baje je bolan. 6) headword: šdua, sl
         example: hudo, de example: schwer zumute sein. In this case, you need to return hudo pri
          šdui." },
  {"role": "user",
    "content": "In some cases, the sl example contains multiple options seperated by a slash or
         similar. For example, headword: drzen, sl example: šala/vic, de example: gewagter Witz.
         In this case, you need to return drzna šala/drzen vic." },
  {"role": "user",
    "content": "In some cases, the sl example contains only an instruction or a description and
         it is up to you to generate the actual phrase. For example, headword: dobiti, sl example:
          pogovorno tudi:, de example: kriegen. In this case, you need to return dobiti." },
  {"role": "user",
    "content": "Pay attention to headwords that start with a dash or hyphen: headword: -komoren,
          sl example: dvo, de example: zweikammerig. In this case, you need to return dvokomoren.
         " }
  {"role": "user",
    "content": f"headword: {g}\nsl example: {sl}\nde example: {de}"}
]
```

Listing 12.7: Prompt used to prime GPT-4o for example-phrase expansion

Two typical tasks handled by the model are shown below in snippet 12.8 and snippet 12.9.

```
<!-- legacy -->
<NUM>002720</NUM><FRB>bazalni</FRB>
<SL>membrana</SL><DE2>Basalmembran</DE2>

<!-- correct output -->
bazalna membrana
```

```
<!-- legacy -->
<NUM>003582</NUM><FRB>zdravilno blato</FRB>
<SL>obloga z zdravilnim blatom</SL><DE2>Fangopackung</DE2>

<!-- correct output -->
obloga z zdravilnim blatom
```

Listing 12.8: Example 1 – example-phrase expansion

In snippet 12.8, the model had to provide the correct translation of the German example `Basalmembran` using the provided Slovenian headword `bazalni` and example `membrana`. Note that in same cases, as indicated in the second example in snippet 12.8, the had to correctly identify that the Slovenian example is already a proper translation of the German example, so no changes were required.

```
<!-- legacy -->
<G>abeceda</G> <BV>imenica</BV> <FRB>klicati po -i</FRB>

<!-- correct output -->
klicati po abecedi
```

Listing 12.9: Example 2 – phrase expansion based on headword

In snippet 12.9, the model had to provide the correct expanded phrase using the provided headword `abeceda` and part of speech `imenica`.

Large Language Models (LLMs) were employed selectively, only for well-defined tasks and under controlled conditions. Budgetary limitations and time constraints precluded comprehensive benchmarking or formal evaluation procedures and instead, the quality of LLM-generated output was assessed through expert review. Experienced lexicographers evaluated the results and exercised their judgment to determine whether the output met the required standards for integration into the transformation workflow. This pragmatic approach allowed us to make targeted use of LLMs where they offered clear benefits, while maintaining overall quality and consistency in the converted data.

5. Wikibase as technical infrastructure for DMLex datasets

5.1 Lexical data in Wikibase

Wikibase,[3] a set of extensions to mediaWiki,[4] provides a platform for storing Linked Data and making it available and collaboratively editable. The platform is generic in the sense that it caters for multiple use cases. At the same time, ensuring interoperability on a basic level, Wikibase's built-in basic data model provides a backbone for the description of ontological concepts (real-world objects and abstract concepts), on the one hand, and of lexical entities, on the other. Concepts (Wikibase *items*) and lexical entities (Wikibase *lexemes*) are linked to each other or to data values using typed *properties*. This allows to create data collections in the shape of a knowledge graph, which can be visualized and queried through the Wikibase interface, where all annotations attached to an entity are shown to the user and can be directly edited; the data can also be accessed

[3] See https://wikiba.se.

[4] See https://mediawiki.org.

programmatically through the mediaWiki API, and through a SPARQL endpoint. Any manual or programmatical edit to a Wikibase entity (*item*, *lexeme* or *property*) is recorded in the entity's edit history for review, and, if regarded necessary, for being reverted. At the same time, any human or bot user's contribution history is also recorded. These features make the Wikibase software an interesting software infrastructure solution for DMLex datasets. The largest collection of ontological and lexical data on an instance of the Wikibase software today is the one at Wikidata.

Lexical entries on a Wikibase (Wikibase *lexemes*) are by default modeled according to the three core classes of the Ontolex-Lemon model (McCrae et al., 2017; Lindemann, 2025), i. e., lexical *Entry*, *Sense*, and *Form*. In addition, a limited set of properties is pre-set in the domain of these three classes, but all other modeling is left to the user. This ensures interoperability of lexeme descriptions regarding the very basics of the description, and, at the same time, leaves space for use-case centered modeling decisions.

The DMLex model for describing dictionary entries is, in principle, compatible to that scheme, since it includes the same core elements in the same hierarchical order; it describes entries, and within those, Sense and Form objects. That is, a migration of DMLex datasets to a Wikibase instance should be straightforward on that basic level. Beyond that, the flexibility of Wikibase should also allow for including all DMLex microstructural elements, and, in addition, provide the infrastructural means for semantic enrichment of the data.

The research question behind the experiments described in the subsequent sections is whether Wikibase turns out a feasible solution for storing DMLex datasets, so that they can be viewed, edited, and enriched by further annotations, and linked to each other, and whether a migration of DMLex datasets to Wikibase can be achieved using a generic method, in the sense that it could be applied to any lexical dataset following the DMLex standard. For the experiments, we used the LexBib Wikibase instance, which was used in a preceding project (Kosem & Lindemann, 2021).[5]

5.2 Remodeling of DMLex sources

The core classes used in a Wikibase to describe lexemes are `ontolex:LexicalEntry`, `ontolex:LexicalSense`, and `ontolex:Form`. On the other hand, the DMLex ontology[6] defines its core classes as subclasses of the Ontolex-Lemon core classes as listed in table 1, so that, for *sense*, a mapping conflict arises. This, in our experiments, has no other solution than using the mapping as shown, i.e, mapping `dmlex:Sense` to `ontolex:Sense` instead of `ontolex:LexicalConcept`. The mapping of *form* classes, in turn, is straightforward.

Regarding the different DMLex microstructural elements, and strategies for how to represent them on Wikibase, it is convenient to first discuss how Wikibase uses the *statement* as standard reification strategy, i. e., as a method for adding qualifying annotations to a semantic triple.[7] For example, a DMLex `<example>` element,[8] which, apart from the exam-

[5] See the documentation of the experiments and pages regarding each dictionary uploads at https://lexbib.elex.is/wiki/DMLex_on_Wikibase.

[6] See https://github.com/oasis-tcs/lexidma/blob/master/dmlex-v1.0/specification/schemas/RDF/dmlex-core.ttl.

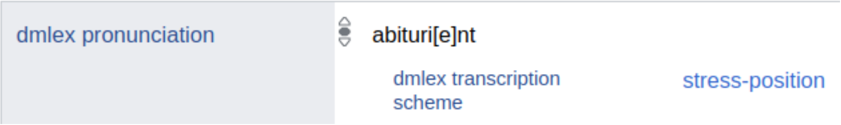
[7] For full details about lexical data modeling in Wikibase, refer to Lindemann (2025).

[8] In these experiments, the DMLex sources to process were serialized in DMLex XML, and our conversion scripts use that as input; consequently, in the presented examples, we refer to the XML serialisation.

DMLex class	Superclass in DMLex ontology	Mapped to in Wikibase
`dmlex:Entry`	`ontolex:LexicalEntry`	`ontolex:LexicalEntry`
`dmlex:Sense`	`ontolex:LexicalConcept`	`ontolex:LexicalSense`
`dmlex:InflectedForm`	`ontolex:Form`	`ontolex:Form`

Table 1: DMLex core classes mappings.

ple text in a `<text>` child element, contains another child named `<exampleTranslation>`, the `<text>` child element of which contains a translation. This, in Wikibase, we represent as qualified statement, as seen in Fig. 1, a screenshot from the Wikibase interface.



dmlex pronunciation

abituri[e]nt

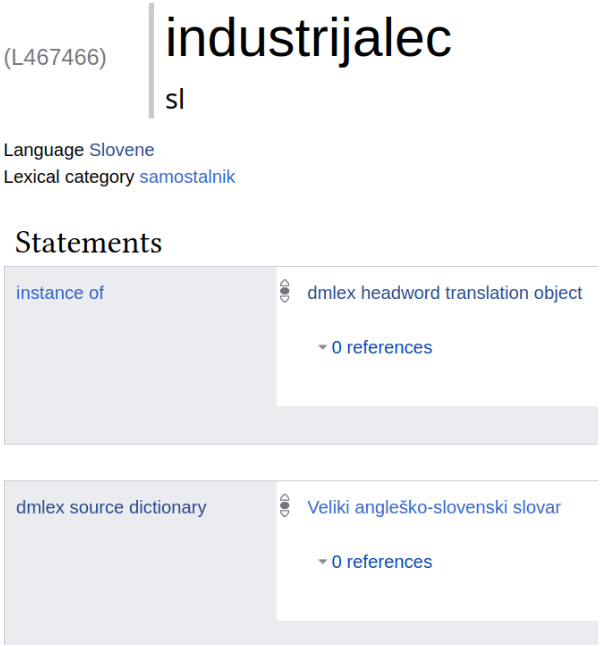
dmlex transcription scheme

stress-position

Figure 1: A qualified Wikibase statement on `lexbib:L467465`.

Multiple qualifiers can be used in the same statement, but no additional reification layer is possible without creating an extra Wikibase *item*, so that qualified statements can be made about that. If following the DMLex model strictly, that is necessary e.g. for `<pronunciation>` elements, since those may contain more than one <transcription> element. Each of the <transcription> elements contain themselves information that would demand a qualified statement on their own, namely the attribute `"scheme"` with a string describing the applied translation scheme as value, together with the transcription `<text>`. Since, in practice, at least in the examples on hand, it not occurs that a `<pronunciation>` element contains more than one `<transcription>`, we have flattened the structure to a simple qualified statement attached to the entry, instead of creating a separate pronunciation object, which would reside, from the user point of view, outside of the lexeme page, and from the database point of view, as separate Wikibase *item*.

This responds to the general interest in keeping all data pertaining to the same lexical entry ”together”, so that it is displayed on the same lexeme entity page. In other cases, this is not possible. For example, the `<headwordTranslation>` element inside `<Sense>` contains a string representing the headword translation equivalent in its `<text>` child, but, in addition, may contain one or more `<inflectedForm>` children. Since that element itself has to be represented as qualified statement (the form string, qualified with the inflected form tag (e.g. *plural*), a separate entity has to be created for the headword translation. We have done so, and we have represented the translation equivalent as Wikibase *lexeme*. That decision forces us to define a language and a lexical category for the lexeme; we infer the former from the general `<translationLanguage>`, and the latter from the lexical category specified at entry level. An example is the Slovenian entry `lexbib:L467466`, ”industrijalec” (see Fig. 2, which originally is listed as headword translation equivalent for a sense in `lexbib:L467465-S1`, the first listed sense in the English-Slovene dictionary with lemma ”industrial”. A sense is created also for the headword translation entry, `lexbib:L467466-S1`,



(L467466) industrijalec

sl

Language Slovene

Lexical category samostalnik

Statements

instance of	dmlex headword translation object 0 references
dmlex source dictionary	Veliki angleško-slovenski slovar 0 references

Figure 2: A headword translation entry, `lexbib:L467466`.

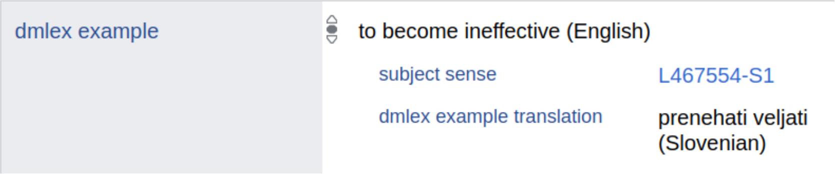
which points back to the translation source sense. An example of a headword translation entry that includes an inflected form is `lexbib:L97074`. The creation of separate entries also has made it necessary to create a new ontology class, not foreseen in the DMLex model, namely `lexbib:Q113`, *headword translation object*. A positive effect of this modeling is that it makes headword translation objects better findable (since lexeme lemmata are indexed in the Wikibase instance's ElasticSearch engine, through the Wikibase lemma search, and also through SPARQL), and more interoperable for alignments with headword lexemes in the dictionary of the contrary direction (here: Slovene-English).

As these two examples show, the original DMLex structure has not been mantained. In the first case, it was flattened, so that, avoiding the creation of a separate entry, all information could be kept in qualified statements, and in the second, the original structure demanded the creation of such separate entry, a lexeme in that case, so that additional information which in the DMLex source is not made explicit had to be included (language and lexical category for the translation equivalent), in order to meet with the minimum requirements for a Wikibase lexeme.

Having such re-modeling decisions in account, we were able to produce a script that converts DMLex XML datasets to a version uploadable to Wikibase, and that is generic in the sense that it processes all datasets used in these experiments in the same way.[9]

We are attaching all microstructural elements found in the source to the corresponding level in a Wikibase lexeme (see mapping in table 1), with one exception: The `<example>` element. Usage examples, according to DMLex, are always attached to `Sense`. In the datasets on hand, however, that requirement has forced the producers to include additional undescribed or ”dummy” `<Sense>` element in the XML as container for those examples that, for some reason, could not be assigned to a described `<Sense>`, that is, to a sense

[9] All scripts are available at https://github.com/dlindem/dmlex-wikibase. The uploaded datasets are accessible at https://lexbib.elex.is/wiki/DMLEX_on_Wikibase.


dmlex example

to become ineffective (English)

subject sense L467554-S1

dmlex example translation prenehati veljati (Slovenian)

Figure 3: An example object at entry level in `lexbib:L467554`, linked to a sense object.

that includes a headword translation or definition. In other words, those undescribed senses contain nothing else than elements of type `<example>`. For a range of reasons we attach examples always to *entry*, not to *sense*, and in case it is assigned to a described sense, qualify it with the corresponding sense ID, as illustrated in the screenshot Fig. 3:

- Wikibase does not allow to create a sense without any textual description. The minimum is a one-word sense gloss in one language.
- When involving sense objects in SPARQL queries, dummy senses would disturb the results.
- The examples not assigned to any of the described senses might, by manual or automatized post-processing, be assigned to a sense. For that task, it is much easier to list examples not linked to any sense than finding those examples part of a dummy sense container, and then, add the qualifier linking to the sense object, instead of deleting and re-writing the example statement inside another sense.
- An example might be good for more than one sense.
- The modeling recommendations on other Wikibases, namely Wikidata, prefer examples being attached to entries and qualify them with the corresponding subject sense or senses (see `wd:P6072`), and, in the same way, qualify it with the corresponding subject *form* (see `wd:P5830`), so that the flexion morphology features of the form used in the example can be specified. This interest rests justification for attaching examples to senses (and not to forms).
- It is a general problem not to be able to assign senses to examples. For example, when collecting example sentences for polysemous headwords from corpora, it is typically not straightforwardly possible to specify the correct sense.

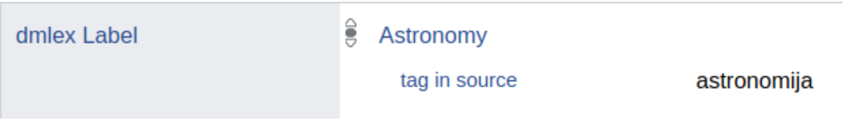
5.3 DMLex and lexical dataset harmonization

As explained in the preceding section, after defining some conversion rules that apply to all datasets used in these experiments in the same way, it has been possible to upload all content. The source datasets used here and potentially in the future are diverse, and it is the function of the conversion to DMLex to harmonize that diversity, so that the Wikibase upload method will be stable throughout all possible use cases. Our experiments suggest that this is possible.

However, we have encountered an issue that is not solvable with rule-based re-modelings as the above explained: According to DMLex, the languages described in the dictionary content are specified at two levels: (1), the language of the lemma list, as `langCode` attribute to the `<lexicographicResorce>` element, which wraps all entries, and (2) the language of the translation equivalents given in the dictionary, as `langCode` attribute to the `<translationLanguage>` element. The value of the former is inherited to all content except the translation equivalents (in DMLex, <headwordTranslation>), which is, in some cases, false. For example, Veliki angleško-slovenski slovar, an English-Slovene dictionary, uses Slovene as metalanguage when providing tags such as those for the lexical category or the domain of knowledge, and definitions, which cannot be represented in the DMLex markup, so that those microstructural items, inheriting the language of the lemma list, are mistakenly treated as English content.[10] In other words, the current DMLex version misses a distinction between the language of the lemma list and the metalanguage of the dictionary. Also, the present modeling of content languages misses a way to encode multiple translation languages, for multilingual dictionaries.

5.4 Controlled values

DMLex describes controlled values as lists of literal values that belong to different tag types, such as the general <labelTag>, or the ones associated to certain microstructure item types like <definitionTypeTag>, <inflectedFormTag>, or <partOfSpeechTag>. For each of these, a Wikibase item is created (see e.g. lexbib:Q34905 ”pridevnik”). These items describing tags remain prepared for being related to each other, most prominently, by declaring items describing the same part of speech throughout dictionaries to be equivalents. The structure of the Wikibase Ecosystem can assist with that task: Existing matching tools[11] can be used to align all items labelled ”pridevnik” to the Wikidata-item describing the *adjective* word class (wd:Q34698).



dmlex Label

Astronomy

tag in source astronomija

Figure 4: A semantically enriched label tag in lexbib:L428040-S4, showing the label in the language of choice of the user.

In the same way, other literal values denoting the same concept but present in the data in different languages or graphical variants (”Adjektiv”, ”adjective”, ”adj.”, etc.) become aligned to the same Wikidata concept. In other words, the literal values present in the DMLex source become enriched with ontological annotations. That allows querying the database content in a more advanced way, and also allows a user to view the tag value in their own language, as in Fig. 4, showing a label statement on lexbib:L428040-S4, where the literal Slovene value ”astronomija” is recorded as a qualifier to the label statement, the main value of which points to a Wikibase item, lexbib:Q34721. That has been annotated with the equivalent Wikidata item identifier (wd:Q333), from where multilingual labels have been imported, and the ontological relations of which (e. g. that Astronomy, according to Wikidata, is a branch of science) now can be included in the retrieval of Wikibase lexemes.

[10] See an example in lexbib:L384053-S1, where the language of the sense definition is marked as English instead of Slovene.

[11] A widely used tool for entity reconciliation tasks is Open Refine, see https://openrefine.org.

6. Conclusions and Outlook

This paper has presented an experimental case study aimed at evaluating the practical usability of the new DMLex standard for converting legacy digitized dictionaries and making them suitable for further processing, specifically import into a structured data environment based on Wikibase. The process involved several stages: the transformation of internally normalized XML to DMLex-compliant format, and the subsequent modeling of this content into RDF triples in accordance with the Wikibase data model, and uploading it to an instance of that software. The goal was not only technical interoperability, but also semantic clarity and reusability of the lexicographic content.

The results of the experiment confirm that DMLex is a highly flexible model that can, in principle, accommodate diverse dictionary structures and contents. Its modular architecture and serialization independence allow it to serve as a bridge between historical lexicographic resources and contemporary digital infrastructures. However, the experiment also revealed some important limitations and challenges that need to be addressed in future work.

A key conceptual issue is the differentiation and representation of three linguistic layers within DMLex: the object language (the language described), the metalanguage (the language used to describe it), and the translation language (used in bilingual or multilingual contexts). In some legacy resources, these layers are not clearly separated or are mixed in ways that complicate structured representation. DMLex offers some mechanisms for this differentiation, but these were not always sufficient for capturing the complexity of real-world data, especially in bilingual and multilingual dictionaries.

Further challenges were identified during the conversion and modeling phases, particularly when the DMLex-encoded data were prepared for import into Wikibase. Several issues, which had not been visible or problematic in the XML serialization, became significant during this phase. Among them were:

- **Inflection suffixes**: Entries in our dictionaries contain tags giving inflection suffixes as indicators of the inflection class the word belongs to; these are treated as `inflectedForm` in the transformation from legacy into DMLex. The result is thousands of entities of type `ontolex:Form` which are all the same.
- **Dummy senses**: Some senses served only as containers for structural organization (usage examples in DMLex need to be placed inside senses) and did not carry semantic descriptions themselves. While this is acceptable in DMLex, it is problematic for Wikibase, where all sense nodes are expected to carry meaning.
- **String languages**: In bilingual dictionaries, elements were sometimes given in a form that lacked correct language tagging, since they by default inherit the language tag associated to the whole resource. Mapping such elements into Wikibase items resulted in false string language declarations.

These findings suggest that while DMLex provides a strong foundation, further specification and modeling guidelines are needed for certain lexicographic phenomena, especially when the data are used outside of traditional lexicographic contexts, such as in linked data environments. In this regard, Wikibase has proven to be a robust infrastructure for presenting, editing, and enriching digitized dictionary content. Towards a generic and replicable method for legacy dictionary LODification, experiences collected in the presented pilot experiments will allow us to refine the whole workflow.

In future work, we plan to expand this pilot to a broader range of dictionaries and test additional features of DMLex, including etymology and annotation modules. Furthermore, the use of language models and other AI-based tools for improving conversion quality, disambiguating ambiguous structures, and enriching dictionary content will be explored more systematically. We also believe that community involvement, including collaboration with dictionary editors and users, will be essential for identifying best practices for the use of DMLex in real-world applications.

In conclusion, this experiment demonstrated both the strengths and the current limitations of DMLex. It is a promising format for standardizing and mobilizing legacy lexical resources, but its successful deployment in complex workflows still requires practical refinement and further alignment with the realities of heterogeneous lexicographic data, on one side, and the practical constraints of Linked Data publishing platforms, on the other.

7. Acknowledgements

This research has been supported by "Hizkuntzalaritza Diakronikoa, Tipologia eta Euskararen Historia / Diachronic Linguistics, Typology and the History of Basque (DLTB)" research group (Basque Government, IT1534-2). The research program Language Resources and Technologies for Slovene (P6-0411) and the projects Adaptive Natural Language Processing with Large Language Models and Large Language Models for Digital Humanities (GC-0002) are funded by the Slovenian Research and Innovation Agency.

Software

Almeida, B., Costa, R., Salgado, A., Ramos, M., Romary, L., Khan, F., Carvalho, S., Khemakhem, M., Silva, R. & Tasovac, T. (2022). Modelling usage information in a legacy dictionary: from TEI Lex-0 to Ontolex-Lemon. In *Workshop on Computational Methods in the Humanities 2022 (COMHUM 2022)*, volume 3602. CEUR Workshop proceedings.

Belyaev, O., Khomchenkova, I., Sinitsyna, J. & Dyachkov, V. (2021). Digitizing print dictionaries using TEI: The Abaev Dictionary Project. In *Proceedings of the Seventh International Workshop on Computational Linguistics of Uralic Languages*. pp. 57–64.

Biffi, M., Sassolini, E., Monachini, M., Montemagni, S. et al. (2019). Converting and structuring a digital historical dictionary of Italian: a case study. In *Electronic lexicography in the 21st century: smart lexicography. Proceedings of the eLex 2019 conference (1-3 October 2019, Sintra, Portugal)*. Lexical Computing CZ, pp. 603–621.

Francopoulo, G. (2013). *LMF lexical markup framework*. John Wiley & Sons.

Kosem, I. & Lindemann, D. (2021). New developments in Elexifinder, a discovery portal for lexicographic literature. In Z. Gavriilidou, L. Mitits & S. Kiosses (eds.) *Lexicography for Inclusion: Proceedings of the 19th EURALEX International Congress, 7-11 September 2021, Alexandroupolis, Vol. 2*. Alexandroupolis: Democritus University of Thrace, pp. 759–766. URL https://euralex2020.gr/proceedings-volume-2/.

Lindemann, D. (2025). Ontolex-Lemon in Wikidata and other Wikibase instances. In *Fifth Ontolex Workshop, September 9, 2025*. Naples: Zenodo. URL https://doi.org/10.5281/zenodo.15471514.

Lindemann, D., Ahmadi, S., Khan, A.F., Mambrini, F., Iurescia, F. & Passarotti, M.C. (2023). When OntoLex Meets Wikibase: Remodeling Use Cases. *CEUR Workshop proceedings*, 2773. URL https://ceur-ws.org/Vol-3640/paper14.pdf.

Maxwell, M. & Bills, A. (2017). Endangered data for endangered languages: Digitizing print dictionaries. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*.

McCrae, J.P., Bosque-Gil, J., Gracia, J., Buitelaar, P. & Cimiano, P. (2017). The OntoLex-Lemon Model: Development and Applications. In I. Kosem, C. Tiberius, M. Jakubíček, J. Kallas, S. Krek & V. Baisa (eds.) *Electronic lexicography in the 21st century: Lexicography from scratch. Proceedings of eLex 2017*. Brno: Lexical Computing CZ s.r.o., pp. 587–597. URL https://elex.link/elex2017/wp-content/uploads/2017/09/paper36.pdf.

Romary, L. & Lopez, P. (2015). Grobid-information extraction from scientific publications. *ERCIM News*, 100.

Salgado, A. (2018). From Legacy Formats and Databases to TEI: Converting the Academy of Sciences Portuguese Dictionary to TEI Lex-0. DigiLex. URL https://doi.org/10.58079/nmo1. Retrieved July 20, 2025.

Tasovac, T., Romary, L., Banski, P., Bowers, J., de Does, J., Depuydt, K., Erjavec, T., Geyken, A., Herold, A., Hildenbrandt, V., Khemakhem, M., Lehečka, B., Petrović, S., Salgado, A. & Witt, A. (2018). TEI Lex-0: A Baseline Encoding for Lexicographic Data. DARIAH Working Group on Lexical Resources. URL https://dariah-eric.github.io/lexicalresources/pages/TEILex0/TEILex0.html.

Tiberius, C., Krek, S., Depuydt, K., Gantar, P., Kallas, J., Kosem, I. & Rundell, M. (2021). Towards the ELEXIS data model: defining a common vocabulary for lexicographic resources. *Electronic lexicography in the 21st century (eLex 2021) Post-editing lexicography*, 91.