Validating the OntoLex-*lemon* Lexicography Module with K Dictionaries' Multilingual Data

Julia Bosque-Gil[1,3], Dorielle Lonke[2], Jorge Gracia[3], Ilan Kernerman[2]

[1] Ontology Engineering Group, Universidad Politécnica de Madrid, Spain
[2] K Dictionaries, Tel Aviv, Israel
[3] Aragon Institute of Engineering Research (I3A), University of Zaragoza, Spain
E-mail: jbosque@funizar.es, dorielle@kdictionaries.com, jogracia@unizar.es, ilan@kdictionaries.com

Abstract

The OntoLex-*lemon* model has gradually acquired the status of *de-facto* standard for the representation of lexical information according to the principles of Linked Data (LD). Exposing the content of lexicographic resources as LD brings both benefits for their easier sharing, discovery, reusability and enrichment at a Web scale, as well as for their internal linking and better reuse of their components. However, with *lemon* being originally devised for the lexicalization of ontologies, a 1:1 mapping between its elements and those of a lexicographic resource is not always attainable. In this paper we report our experience of validating the new *lexicog* module of OntoLex-*lemon*, which aims at paving the way to bridge those gaps. To that end, we have applied the module to represent lexicographic data coming from the Global multilingual series of K Dictionaries (KD) as a real use case scenario of this module. Attention is drawn to the structures and annotations that lead to modelling challenges, the ways the *lexicog* module tackles them, and where this modelling phase stands as regards the conversion process and design decisions for KD's Global series.

Keywords: Linguistic Linked Data; RDF; multilingual; OntoLex-*lemon*; K Dictionaries

1. Introduction

Linked data (LD) technologies are increasingly adopted in lexicography, whether in academic research and development, the industry, or combining both (see for instance Klimek and Brümmer (2015), Declerck et al. (2015), Abromeit et al. (2016), Parvizi et al. (2016), Bosque-Gil et al. (2016a) and Kaltenböck & Kernerman (2017)). LD refers to a set of best practices for exposing, sharing and connecting data on the Web (Bizer et al., 2009). The adoption of LD in lexicography enhances the tendency to standardize the ways of representation and query of lexical content at a Web scale. Connections can also be established to other LD resources, so that lexicographic data can be enriched with different types of complementary information, such as additional translations, definitions, examples of usage, etc.

The *de-facto* standard for representing ontology lexica, the *lemon* model (McCrae et al., 2012) and its more recent version, OntoLex-*lemon*[1] (McCrae et al., 2017), have been the preferred choice by developers to convert lexicographic resources into LD. Early experiences in using *lemon* show that the model is highly effective as regards the accounting for the core lexical information in lexicographic resources (Klimek & Brümmer, 2015; Declerck et al., 2015; Abromeit et al., 2016; McCrae et al., 2019). However, there are various situations in which no perfect match is available between the elements of the model and those found in lexicographic entries, or in which the model falls short of capturing certain peculiarities of lexicographic works, e.g. the order of senses in an entry, details on the morphological features of word-forms when used for a specific sense, etc. In this context, the W3C OntoLex community group[2] has analysed the main issues regarding the representation of lexicographic information as LD and is releasing this year an updated module to represent lexicographic data that extends the *lemon* core model – the *lexicog* module.[3]

In this paper we analyse the application of the *lexicog* module for LD-based representation of the Global series of K Dictionaries (KD).[4] The main contribution of this work is twofold:

1. This pioneering experience serves to validate this new module with an actual use case as well as to introduce some recommendations for future applications.

2. By focusing on KD's data, we examine how the limitations of the OntoLex-*lemon* model already reported in the literature (Klimek & Brümmer, 2015; Bosque-Gil et al., 2016b) are successfully addressed by the module.

The rest of this paper is structured as follows: Section 2 provides an overview of KD's Global series and elaborates on the motivation for its conversion to LD, as well as a summary of previous conversions of these data to LD and the challenges encountered in this process. In Section 3 the *lexicog* module is introduced. Section 4 briefly presents the different stages of LD generation, and where the modelling with *lexicog* stands with respect to the whole conversion of KD's data to the Resource Description Framework (RDF), along with the technologies and the design decisions we adopted. Section 5 addresses some of the limitations previously detected in the literature on the conversion of KD's data and provides a modelling solution in terms of *lexicog*. Concluding remarks and future lines of work are presented in Section 6.

[1] https://www.w3.org/2016/05/ontolex/

[2] https://www.w3.org/community/ontolex/

[3] The *lexicog* module and report are available at at http://www.w3.org/ns/lemon/lexicog# and http://www.w3.org/2019/09/lexicog/ respectively.

[4] http://www.lexicala.com/.

2. K Dictionaries' Global series

In this section we briefly describe the dictionary data that we used to validate the *lexicog* module, which stems from the Global series of KD. This series is based on the monolingual lexicographic cores of 25 different languages and their bilingual and multilingual versions, and includes nearly 100 language pairs and numerous multilingual variations. We discuss the motivation of converting it into LD and describe preliminary conversions that were performed in the past.

2.1 Converting KD's data to the RDF: motivation and overview

The Global series of KD (Kernerman, 2009, 2011, 2015) has been conceived as a cross-lingual, multi-layer mosaic of lexicographic resources that evolve within a single systemic framework, sharing a common technical macrostructure and a common entry microstructure that is able to accommodate and adapt to particular characteristics of different languages. All the language sets share the same XML schema (DTD), wherein certain languages can feature additional orthographic scripts (e.g., have Kanji, Hiragana, Katakana and Romaji for Japanese, or encompass diverse inflected verb forms, for example, perfective/imperfective for Polish and Russian). Each language resource is created on its own, based on deep corpus analysis from which stem its editorial style guide, headword list, lexical deciphering and mapping, and diverse semantic and syntactic attributes. The result is a detailed monolingual core that might contain overlapping elements, such as definitions alongside sense disambiguation elements, synonyms or antonyms, etc., which can then be used selectively to customize that data to the needs of particular target audiences and usages. This core is ready to be complemented by translation equivalents (of the senses, examples of usage, and multiword units) for developing bilingual versions, which are juxtaposed and form a multilingual network revolving around the initial monolingual set. Eventually, the translations (and other components) of each language network can also be interlinked to each other and exponentially multiply the cross-lingual connections.

Since its inception in 2005, 25 language cores were created, and altogether nearly 100 language pairs are available so far, besides numerous multilingual combinations. Rather than aim to compile any specific dictionary product, the idea was to develop multifunctional data sets that can be applied in different forms and media, either independently or in conjunction with other data, whether intended to publish a print dictionary, develop an online or a mobile dictionary, offer lexical services, or be incorporated in NLP applications. The advent in recent years of Linguistic LD and Semantic Web technologies has opened new horizons to enhance this strategic approach of creating well-structured, detailed and extensive lexicographic data rather than single dictionary products, by reinforcing and further expanding existing data, and improving interoperability between content from the Global series and other multilingual data on the Web, attaining reciprocal enrichment of the Global series by external resources (on

the one hand), and enhanced incorporation of data from the Global resources into external ones (on the other hand). To put this notion into practice it became necessary to first transform KD's Global data from its original XML (hierarchical) format to an RDF structure (knowledge graph), for smoother linking to external resources. Thus, KD decided to apply the best-known LD standard model for representing lexicographic content, first in the form of *lemon*, then conforming to OntoLex-*lemon*, and most recently in line with its up-to-date *lexicog* module.

The motivation of KD to focus and invest in this venture can thus be explained by the invaluable upgrade this should carry for its resources, through facilitating their interoperability and enhancing depth, precision, and cross-linguality. Such improved features are needed to deal with the emerging multilingual single digital market, primarily in Europe and eventually all over the world, which calls for multiple adaptations of content and technology, international standards, multi-disciplinarity, etc. LD methods are at the forefront of the current generation of powerful language technology solutions, at the heart of human-machine interaction. Providing quality cross-lingual lexical data, with the LD-driven option of linking to other sources, greatly increases the appeal and uniqueness of the KD resources and places KD in a leveraged position to other competing dictionary APIs.

The new API of KD, renamed Lexicala API, provides access to the Global (and other KD) data in JSON, with the first touches of JSON-LD. It constitutes a vital step in an innovative trend of turning passive dictionary products into active lexical data services that interoperate with real-world computational linguistics applications. Two ongoing H2020 projects employ Lexicala API as part of their solutions: Lynx[5] will integrate KD (as well as terminological and other) resources with data from the legal domain in the heart of its Legal Knowledge Graph platform for multilingual compliance services; and Elexis[6] will make use of the API to receive KD content for its future European lexicographic infrastructure. Making KD resources available in state-of-the-art RDF conforming to world-class standards will both help to enhance the operation of Lynx and Elexis platforms, and those of a multitude of future applications, and to reinforce and expand KD content through interaction with more LD resources.

2.2 Previous representations of KD's data as RDF

The current conversion of KD's multilingual Global series is not the first effort to convert this data to RDF. In 2014 KD became involved in the first attempt to convert Global data from XML format to RDF, adhering to the *lemon* model and focusing on the German monolingual dataset (Klimek & Brümmer, 2015). The next massive step was taken in the two-year project carried out in 2015-2017 as part of a EUREKA

[5] http://lynx-project.eu/

[6] https://elex.is/

bilateral framework between KD and Semantic Web Company (SWC), called Linked Data Lexicography for High-End Language Technology Application (LDL4HELTA).[7] As part of the LDL4HELTA project, the Global data for three languages (English, German and Spanish) was converted to RDF in line with the OntoLex-*lemon* model (Bosque-Gil et al., 2016b).

In the first work (Klimek & Brümmer, 2015), the authors identified some gaps in the *lemon* model with regard to representing KD's data, for instance, the way to link a compound phrase defined inside of a sense group to that same sense. The lack of an ontology to provide ontological references for lemon:LexicalSenses was also highlighted. This point is strongly connected to the original aim of the *lemon* model to serve to lexicalize ontologies, not to represent lexicographic resources in the Web of Data. In addition, the authors identified some gaps in the LexInfo[8] catalogue of grammar categories (typically used in conjunction with *lemon*) and created their own custom vocabulary to capture the values of KD's DTD attributes. In the later conversion of the series to OntoLex-*lemon* (Bosque-Gil et al., 2016b), some problems that were identified in the previous conversion were no longer relevant, as both the model and its modules had evolved to cover more cases (e.g. now the *vartrans* module allows to represent lexical relations).

It is worth noting that, whereas in the first two attempts the conversion was carried out under the strict principle of round-tripping, i.e. aiming to obtain full and complete 1:1 data transformation from XML to RDF and from RDF back to XML – so the RDF structure had to convey each and every detail of the complex features of the original XML structure – the current work was released from this obligation. The reasons for applying such a demand in the first place were, on the one hand, to serve as validation of perfect transfer from XML to RDF while, on the other hand, to be able to benefit from the potential enrichment of the data in RDF when linking to other data resources and importing such new data back to the existing resource in XML. Removing this restriction has helped to liberate and enhance the data flow from one format to another, and emphasized the autonomous status of each model and the fact that every format should behave freely and reflect its autonomous characteristics that are different from the other.

However, OntoLex-*lemon* proved to be not exhaustive enough to cover the representation requirements of the original resource. Four kinds of challenging situations were detected in the modelling of KD's multilingual data:

1. Cases in which solely applying the OntoLex-*lemon* model would lead to a loss of structural information reflecting lexical distinctions. For example, entries *not* originally conceived as dictionary entries in KD data are treated equally as

[7] https://www.eurekanetwork.org/project/id/9898

[8] https://lexinfo.net/ontology/2.0/lexinfo.owl

original entries in the RDF representation (i.e. as ontolex:LexicalEntry). This highlighted a lack of elements for representing the components of a lexicographic entry in cases in which there is no 1:1 mapping with OntoLex-*lemon* classes and properties. In KD data, we encounter several examples of this type of situation: compounds, synonyms, antonyms, and translations. Compounds are defined inside the dictionary entry as one of its components and do not occur as lemmas (i.e. in their own dictionary entry). Synonyms and antonyms, even though they are usually independent lemmas in that same resource, are embedded in dictionary entries and they do not necessarily have their corresponding dictionary entry in that resource (but could occur as dictionary entries in another KD dictionary). In addition, a translation of a headword into another language is treated as an ontolex:LexicalEntry (Bosque-Gil et al., 2016b), too, but just as a synonym, and the source data in its current state does not guarantee for the word to be a lemma in the dictionary of the target language. This fact called for a distinction between an original dictionary entry and the ontolex:LexicalEntry newly created in the process, thus recording the outcome of the headword selection step in the compilation of the dictionary. In lexicographic resources other than KD, the same gap would surface in those cases in which a dictionary entry needs to be split into more than one ontolex:LexicalEntry, each with a different part of speech, in order to be OntoLex-compliant.

2. Cases in which OntoLex-*lemon* or LexInfo falls short of covering the representation needs that KD's dictionary entries give rise to. This concerned the representation of examples and translations of examples, which are fairly common elements in other dictionaries as well (Bosque-Gil et al., 2017).

3. Cases in which OntoLex-*lemon* does contain elements to cover a particular type of information, but there are no specific guidelines on how to use them in the process of conversion of lexicographic data to RDF (without involving ontology lexicalization). For example, the representation of lexicographic definitions with the OntoLex core (e.g. with skos:Concept or ontolex:LexicalConcept), the encoding of geographical usage restrictions on senses, or the modelling of selectional restrictions for predicate arguments.

4. Mismatches between LexInfo elements and KD's DTD tags and values.

Since situations of types (1) and (2) were also generalizable to other lexicographic resources, *lexicog* was proposed as an extension of OntoLex-*lemon* (Bosque-Gil et al., 2017). For cases of type (3), the OntoLex Community, in its bi-weekly telcos on lexicography, discussed a series of practices for the use of OntoLex-*lemon* elements in the conversion of lexicographic data to RDF.[9] These practices emerged as solutions to

[9] https://www.w3.org/community/ontolex/wiki/Lexicography

a list of issues detected in the literature. A series of guidelines, with more examples and recommendations, are also planned as future steps in the OntoLex community. Cases of type (4) were addressed in 2016 by creating a custom ontology for KD, which is currently under revision and update.

3. The *lemon* lexicography module: *lexicog*

The *lemon* model has been extensively used for representing lexicographic data. However, some limitations were detected in several preliminary experiences, as reported in Section 1.

Such issues were collected and analysed by the W3C OntoLex community group with the aim of reaching some agreement that allows for a better and more interoperable migration of existing dictionaries into LD. As a result of this community effort, the Ontolex-*lemon* lexicography module (*lexicog*) was defined as an extension of the OntoLex-*lemon* model.[10] The module is targeted at the representation of dictionaries and any other linguistic resource containing lexicographic data, and addresses structures and annotations commonly found in lexicography.

The *lexicog* module overcomes some limitations of OntoLex-*lemon* when modelling lexicographic information as LD. It aims at capturing the underlying original structure and annotations of the lexicographic entry in a way that keeps the purely lexical content separate from the lexicographic one, minimizing information loss and allowing queries restricted to the lexical layer. By being able to keep record of the original dictionary arrangement as RDF, the module does not impose a certain view on the lexicon and thus becomes agnostic to the standpoint of the lexicographer. For that purpose, new ontology elements have been added that reflect the dictionary structure (e.g., sense ordering, entry hierarchies, etc.) and complement the OntoLex-*lemon* lexicon.

Figure 1 depicts the main classes and relations defined in the *lexicog* module. We refer to the specification document for more details, but we give here an overview of its main modelling ingredients:

- LexicographicResource, which represents a collection of lexicographic entries in accord with the lexicographic criteria followed in the development of that resource.

- Entry, a structural element that represents a lexicographic article or record as it is arranged in a source lexicographic resource.

- LexicographicComponent, which is a structural element aimed at representing the (sub)structures of lexicographic articles providing information about entries,

[10] A record of the discussed issues and intermediate design decisions can be found at https://www.w3.org/community/ontolex/wiki/Lexicography.

senses or subentries. Lexicographic components can be arranged in a specific order and/or hierarchy.

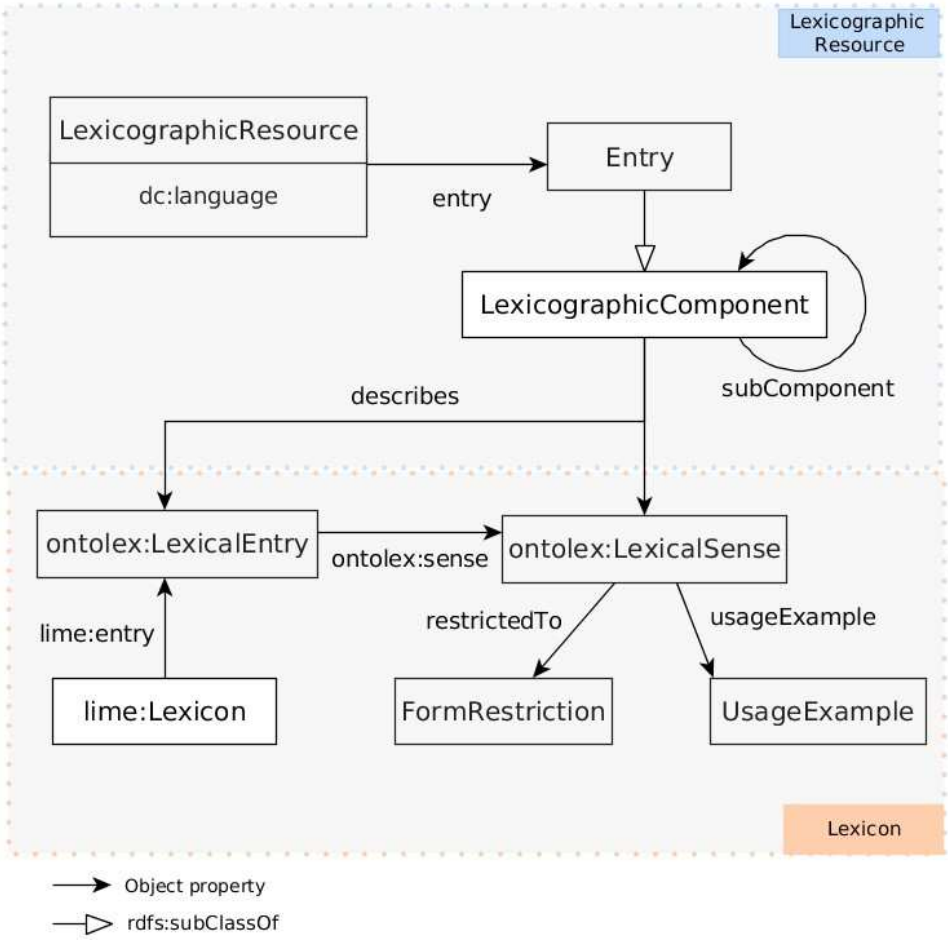

Figure 1: Scheme of the lexicography module (taken from the "OntoLex-*lemon* Lexicography Module" W3C community group final report).

The three above elements account for the basic structure of the LexicographicResource. To that end, a property entry relates a LexicographicResource to each of its entries. An Entry in turn can group several LexicographicComponents. We can indicate that the components belong to an entry by simply using the RDF native mechanisms for containers.[11] In particular, the rdfs:member property can be used if the order of the components is not relevant, and rdfs:ContainerMembershipProperty (rdf:_1, rdf:_2, ...) when the order of the components needs to be represented. Notice that an Entry is a particular subclass of LexicographicComponent used to represent the main "entry point" in the dictionary, i.e., the headword or the root of the lexicographic record.

The lexicographic components only reflect the structure of the dictionary and do not encode any lexical content themselves. To associate them to their corresponding lexical information (e.g. lexical entries or lexical senses), the property "describes" is used. Such

[11] https://www.w3.org/TR/rdf-schema/#ch_containervocab

elements belonging to a lexicon are taken from OntoLex, in particular:

- ontolex:LexicalEntry, which consists of a set of forms that are grammatically related and a set of base meanings that are associated with all of these forms.
- ontolex:LexicalSense, which represents the lexical meaning of a lexical entry when interpreted as referring to the corresponding ontology element.
- lime:Lexicon, or a collection of lexical entries for a particular language or domain.

These classes can be further connected with many other elements that describe the lexicon and that can be found in the OntoLex-*lemon* specification. Particularly, the ontolex:Form class, to account for the grammatical realization of a lexical entry (typically by means of its written representation) and the ontolex:LexicalConcept class, that can be used to store definitions through the property skos:definition.

Finally, we mention the UsageExample class of the *lexicog* module, which is intended to represent a textual example of the usage of a sense in a given lexicographic record.

4. Methodology

4.1 Incremental approach and steps taken

The process of converting KD data into LD was carried out with an incremental approach, starting with the very basics of a single entry (headword, senses, part of speech, definitions) and proceeding with more complicated elements (synonyms, translations, examples of use, compounds, etc.), validating the results of the conversion after each iteration. This approach allowed for constant validation and error elimination, and facilitated the technical conversion process. Prior to converting actual data, some groundwork was necessary. For this purpose, the DTD of KD's XML data was examined, and each XML path in KD data was manually defined as a corresponding OntoLex, *lexicog* or LexInfo element. Next, a URI naming strategy was established, following the previous conversion of the Global series (Bosque-Gil et al., 2016b). In addition, the DTD was revised and edited where possible, adhering to the standards set by LexInfo and OntoLex and prioritizing smooth conversion and adaptable results.

After setting the foundations for conversion, the following steps were taken for each iteration:

- Identifying a few entities in *lexicog* to test, and manually creating an example RDF entry with real KD data. Only a handful of components comprising a complete dictionary entry were selected for each iteration, to simplify each step and govern the results more easily. In order to maintain that the conversion was carried out exhaustively and accurately, logs were kept, and the URI naming

strategy was under constant revision and scrutiny.

- Writing and running a conversion script. The manually constructed example had a vital part in determining the conversion script. The RDF conversion pipeline relies on already existing conversion of XML data into JSON, adding LD elements and restructuring the JSON document to comply to the triple relations encompassed in the JSON-LD structure. In each iteration, the conversion was applied to all of the resources of the Global series, resulting in a collection of JSON-LD documents, with each dictionary entry represented by its own JSON document and reflecting an RDF graph introducing only the components that were the focus of the current iteration, on top of previously covered components.

- Validating output RDF. The final step for each iteration was validating the results.

- The method of validation selected to this end is twofold, consisting of the JSON Schema as an initial means of validation, and a SPARQL endpoint and query service for querying the RDF output.

- Repeat for the next components.

These steps allowed for constant appraisal and control. Further iterations were conducted with taking into consideration any conclusions drawn on their predecessors, and the workflow enabled simultaneous work on the theoretical conversion alongside writing the conversion script by all team members. In particular the JSON schema was very important, as this provided exhaustive validation as part of the pipeline prior to the querying phase.

4.2 Performing the validation

The validation process consists of two parts: the first, initial validation is conducted by defining a JSON schema and validating the JSON-LD documents against it as part of the conversion pipeline; the second, final validation is uploading the RDF output onto a SPARQL query service, e.g. any triple store supporting JSON-LD, and querying the data to certify that all of the input data was properly converted.

The selection of JSON schema for initial validation of the JSON-LD documents was a natural one; designed to validate JSON documents, the schema can be tailored to specific needs and ensure that the JSON document is well-structured and includes only desired elements. The same principles can be applied to JSON-LD, harnessing the advantages of JSON schema to control the triple structure and ensure that URIs are well-defined. The main points of validation offered by the schema are the following:

1. The JSON schema checks that the predicates are in place, that is, that there will not be a JSON object nested inside another JSON object where no relation

exists between them. Together with the context, which can be validated both manually and automatically, the schema basically checks that the correct triple relations occur, and that there are no relations that should not occur.

2. It checks that all necessary information is present, and that nothing was left out during conversion.

3. It checks that the JSON does not contain anything that should not be there, insofar that if something is not specified in the schema but appears in the document, it constitutes an error.

4. It checks that the URIs are well-defined by defining regular expressions according to the URI naming pattern.

By checking these four points, the schema corroborates both the triple relations and the URIs, essentially providing complete structural validation. A JSON-LD document that validates against the JSON schema is trusted to represent a correct RDF graph. Including JSON schema as part of the conversion pipeline ensures that the RDF output is valid, adding another layer of security prior to the querying phase, and establishing that the data stored on the triple store is well-structured and complete.

The chosen serialization, JSON-LD, was selected due to it being a standard and widely used format for structured data among the target sector of API users. Its native compatibility with JavaScript allows for flexibility and customization when converting proprietary data. Its inherently nested structure prevents redundancy and verbosity, and being the main format for API responses it can be easily parsed and manipulated. Furthermore, by defining clear and intuitive aliases for RDF classes, properties and predicates, it has the advantage of being human, as well as machine readable.

The JSON schema, while applicable only to the JSON-LD serialization, encompasses all of the relevant principles of RDF validation, which can be derived directly and applied to any other means of validation used for validating other serializations.

5. Applying *lexicog* to KD's multilingual data

The *lexicog* module draws a distinction between the lexical layer, captured mainly by OntoLex, and the structural elements that describe the lexicon and can be arranged as desired in a particular lexicographic work. We will adhere to this distinction in this section as well and first present problematic cases of KD of type (1) (see Section 2.2), concerning the distinction between a dictionary entry and an ontolex:LexicalEntry and the grouping of dictionary entries, and will follow with the representation of examples and their translations as LD.

5.1 lexicog:Entry and ontolex:LexicalEntry

One of the shortcomings of OntoLex-*lemon* concerned the lack of a way to capture what was originally a dictionary entry in the resource and differentiate it from an ontolex:LexicalEntry created on the fly during the conversion process, which may or may not have their corresponding dictionary entry in the resource (or in a work of the same series, i.e. the Global series from KD). In addition, a lime:Lexicon gathers a collection of ontolex:LexicalEntry elements, which, in turn, can share the language and come from different lexicographic resources from the same series (see Gracia et al., 2018). The lime:Lexicon class is thus not intended to uniquely represent the lexicographic resource as it was conceived originally, but as a collection of lexical entries belonging to the lexicon of a language.

In KD's data, compounds, synonyms, antonyms and translations are defined or described inside a dictionary entry of another lemma (in the case of compounds, inside the dictionary entry of one of their components). In order to represent their definition, form, inflection or pronunciation according to the OntoLex core, they need to be treated as ontolex:LexicalEntry elements, which causes the distinction between original dictionary entries and embedded lexical entries to be lost.

Example 1.1 shows an extract of the dictionary entry *arte* 'art' in Spanish, with its translation into Dutch and the definition of the compound *artes plásticas* 'visual, plastic arts'. This example, in addition to a description of the headword (shortened due to space constraints) provides a synonym in its first sense (*inspiración* 'inspiration'). Below the section devoted to translations, the compound *artes plásticas* is defined.

By applying *lexicog* to example 1.1, we instantiate different elements to represent lexical entries and dictionary entries respectively. Example 1.2 shows an extract of the RDF Turtle representation of example 1.1. The elements in blue refer to the lines in the RDF that mark this distinction. While the Spanish and Dutch lexica gather any unit of the lexicon that is described in the original dictionary (as a dictionary entry or as an embedded entry), represented as ontolex:LexicalEntry, a lexicog:LexicographicResource is intended to group only dictionary entries through lexicog:Entry. This way, the RDF reflects that *artes plásticas* is a unit of the lexicon but it is not a lemma in this dictionary.

lexicog:Entry serves a structural function to only capture the structure of the resource as a result of the lexicographic selection process, and it does not bear lexical information. To close this gap, the property lexicog:describes links dictionary entries (as structures) to the lexical units in the lexicon. If the RDF representation were also to reflect that *artes plásticas* or *inspiración* are lexical entries "defined" inside the dictionary entry of *arte*, the *lexicog* module would provide elements to establish this structural connection. In this case, however, reflecting the whole microstructure of the entry was not a requisite for the expected output; we limit ourselves to capture the semantic relations

between these different lexical entries (translation, synonymy) through the elements of the OntoLex-*lemon* model, following previous approaches (Bosque-Gil et al., 2016b) based on the *vartrans* module.

```
<DictionaryEntry identifier="DE00005536" version="1">
 <HeadwordCtn>
   <Headword>arte</Headword> [...]
 </HeadwordCtn>
   <SenseBlock>
     <SenseGrp [...]>
       <Synonym>inspiración</Synonym> [...]
       <TranslationCluster [...]>
        <Locale lang="nl">
         <TranslationBlock>
           <TranslationCtn>
             <Translation>kunst</Translation> [...]
           </TranslationCtn>
         </TranslationBlock>
        </Locale> [...]
         </TranslationBlock>
       </TranslationCluster>
      <CompositionalPhraseCtn version="1"> [...]
       <CompositionalPhrase>artes
        plásticas</CompositionalPhrase> [...]
      </CompositionalPhraseCtn> [...]
     </SenseGrp> [...]
   </SenseBlock>
 </DictionaryEntry>
```

Example 1.1: An extract of the dictionary entry *arte* 'art' in Spanish from KD's Global series with its translation into Dutch and the compound *artes plásticas* 'visual plastic arts'.

```
@prefix base: <http://lexicala.com/id/global/> .
@prefix lime: <http://www.w3.org/ns/lemon/lime#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix lexicog: <http://www.w3.org/ns/lemon/lexicog#> .
@prefix ontolex: <http://www.w3.org/ns/lemon/ontolex#> .

:mlds-ES3 a lexicog:LexicographicResource;
  dc:language "es" ;
   lexicog:entry :ES_DE00005536 .
```

```
:ES_DE00005536 a lexicog:Entry ;
   lexicog:describes :lexiconES/arte-n .

:lexiconES/arte-n a ontolex:LexicalEntry .
:lexiconES/artes-plásticas-n a ontolex:LexicalEntry .
:lexiconES/inspiración-n a ontolex:LexicalEntry .
:lexiconNL/kunst-n a ontolex:LexicalEntry .

:lexiconES a lime:Lexicon; lime:language "es" ; lime:entry :lexiconES/arte-
  n, :lexiconES/artes-plásticas-n, :lexiconES/inspiración-n .

:lexiconNL a lime:Lexicon;
  lime:language "nl";
  lime:entry :lexiconNL/k
  unst-n.
```

Example 1.2: RDF Turtle representation of example 1.1

5.2 Nested entries

There are other types of information in KD's Global series that require the RDF version of the dictionary to reflect structural aspects. In KD's DTD, the element NestEntry works as a container grouping together several dictionary entries. Example 1.3 in XML shows the entry of the verb *besuchen* 'to visit' in German. The element NestEntry groups together three different dictionary entries: *besuchen* (v. 'visit'), *Besuch* (n. 'visit') and *Besucher* (n. 'guest, visitor') that are related, although the nature of relation is not *explicitly* stated. These containers group together derivations or, in some cases, verbs that share a lemma but not the subcategorization value and are not homonyms.

Example 1.4 shows the RDF rendering of example 1.3 in Turtle serialization. In *lexicog*, grouping is reflected by creating a lexicog:LexicographicComponent and indicating that other components, namely, the three dictionary entries *besuchen*, *Besuch* and *Besucher* (as lexicog:Entry elements) are contained in that component. This is captured by the property rdfs:member.

```
<Entry HomNum="" hw="besuchen" identifier="EN00002666" pos="verb">
    <NestEntry>
      <DictionaryEntry identifier="DE00003297" version="1">
        <HeadwordCtn>
          <Headword>besuchen</Headword> [...]
        </HeadwordCtn> [...]
      </DictionaryEntry>
      <DictionaryEntry identifier="DE00003298" version="1">
        <HeadwordCtn>
```

```
        <Headword>Besuch</Headword> [...]
      </HeadwordCtn>
    [...]
    </DictionaryEntry>
    <DictionaryEntry identifier="DE00003299" version="1">
      <HeadwordBlock>
        <HeadwordCtn>
         <Headword>Besucher</Headword> [...]
        </HeadwordCtn>
        [...]
      </HeadwordBlock>[...]
    </DictionaryEntry>
  </NestEntry>
</Entry>
```

Example 1.3: An extract of the German entry *besuchen* 'visit' with a NestEntry container that groups the dictionary entries *Besuch* 'n. visit' and *Besucher* 'guest, visitor'.

(Continuation)

```
:lexiconDE/besuchen-v a ontolex:LexicalEntry .
:lexiconDE/Besuch-n a ontolex:LexicalEntry .
:lexiconDE/Besucher-n a ontolex:LexicalEntry .

:lexiconDE a lime:Lexicon; lime:entry :lexiconDE/besuchen-
  v, :lexiconDE/Besuch-n, :lexiconDE/Besucher-n.

:mlds-ES3
lexicog:entry :DE_DE00003297, :DE_DE00003298, :DE_DE00003299 .

:DE_DE00003297 a lexicog:Entry;
  lexicog:describes :lexiconDE/besuchen
  -v .

:DE_DE00003298 a lexicog:Entry ;
  lexicog:describes :lexiconDE/Besuch-n.

:DE_DE00003299 a lexicog:Entry ;
  lexicog:describes :lexiconDE/Besucher-
  n .

:DE_EN00002666 a lexicog:LexicographicComponent ;
 rdfs:member :DE_DE00003297, :DE_DE00003298, :DE_DE00003299 .
```

Example 1.4: RDF rendering of the NestEntry structure presented in example 1.3 in Turtle serialization

5.3 Usage Examples

The data from KD's Global series provides, for each sense of a headword, a usage example in the source language and the translations of the headword in the target language. The examples, in turn, are also translated to the target language and serve as example of usage for the translation. Example 1.5 shows another excerpt of the entry *arte* in Spanish. Inside the SenseGrp encapsulating the information of the first sense, there is an element TranslationCluster with Locale groups that include the headword translations for *arte* in its first sense: *kunst* (Dutch) and *kunst* (Norwegian). Below the translations follows an ExampleCtn with the example of usage of *arte* in that sense, *La música, la danza y la pintura son formas de arte* 'Music, dance and painting are art forms'. This example is in turn translated to Dutch and Norwegian.

```
<SenseGrp identifier="SE00007455" version="1">
  [...]
  <TranslationCluster identifier="TC00017354" text="manifestación humana con intención
      estética" type="def">
    <Locale lang="nl">
      <TranslationBlock>
        <TranslationCtn>
          <Translation>kunst</Translation> [...]
        </TranslationCtn>
      </TranslationBlock>
    </Locale>
    <Locale lang="no">
      <TranslationBlock>
        <TranslationCtn>
          <Translation>kunst</Translation> [...]
        </TranslationCtn>
      </TranslationBlock>
    </Locale> [...]
  </TranslationCluster>
  <ExampleCtn type="sid" version="1">
    <Example>La música, la danza y la pintura son formas de
    arte.</Example>
    <TranslationCluster identifier="TC00017355" [...]>
      <Locale lang="nl">
        <TranslationBlock>
          <TranslationCtn>
            <Translation>Muziek, dans en schilderen zijn vormen van kunst.</Translation>
          </TranslationCtn>
        </TranslationBlock>
```

```
        </Locale>
        <Locale lang="no">
          <TranslationBlock>
            <TranslationCtn>
              <Translation>Musikk, dans og maling er kunst
            typer.</Translation> </TranslationCtn>
          </TranslationBlock>
        </Locale> [...]
      </TranslationCluster>
    </ExampleCtn>
  </SenseGrp>
```

Example 1.5: An extract of the Spanish entry *arte* with translations into Dutch and Norwegian examples and translations of the examples.

While the *lemon* model provided a class lemon:UsageExample and a property lemon:example, used previously in the literature to capture this information (Klimek & Brümmer, 2015), these are no longer included in the OntoLex-*lemon* model. Previous conversions of KD's data (Bosque-Gil et al., 2016b) proposed a custom class in order not to instantiate both *lemon* and OntoLex-*lemon* in the same resource. If an example is to be linked to a sense, the property skos:example would suffice to include the example as a string at the sense level. For cases in which the example has additional information, or has elements linkable to it, the *lexicog* modules offers the class lexicog:UsageExample to link an ontolex:LexicalSense to an element representing the example. A lexicog:UsageExaple and be further linked to other elements and described with data-type properties.

Example 1.6 shows the RDF Turtle representation of example 1.5. As showed in example 1.2, the headword and the translations belong to different lexica, one per language.

(Continuation)

```
:lexiconES/arte-n a ontolex:LexicalEntry ;
  ontolex:sense :lexiconES/arte-n-SE00007455-
  sense .
:lexiconNL/kunst-n a ontolex:LexicalEntry;
  ontolex:sense :lexiconNL/kunst-n-arte-n-
  SE00007455-sense .
:lexiconNO/kunst-n a ontolex:LexicalSense;
  ontolex:sense :lexiconNO/kunst-n-arte-n-
  SE00007455-sense .

:lexiconNO a
  lime:Lexicon;
  lime:language "no";
```

```
  lime:entry :lexiconNO/
  kunst-n .

:lexiconES/arte-n-SE00007455-sense a ontolex:LexicalSense ;
  lexicog:usageExample :lexiconES/arte-n-SE00007455-sense-
  TC00017355-ex .
:lexiconNL/kunst-n-arte-n-SE00007455-sense a ontolex:LexicalSense ;
   lexicog:usageExample :lexiconES/arte-n-SE00007455-sense-
   TC00017355-ex .
:lexiconNO/kunst-n-arte-n-SE00007455-sense a ontolex:LexicalSense ;
  lexicog:usageExample :lexiconES/arte-n-SE00007455-sense-
  TC00017355-ex .

:tranSetES-NL/arte-n-SE00007455-sense-kunst-n-arte-n-SE00007455-sense-TC00017354-trans
      a vartrans:Translation ;
     vartrans:source :lexiconES/arte-n-SE00007455-sense;
     vartrans:target :lexiconNL/kunst-n-arte-n-SE00007455-
     sense; dc:source :mlds-ES3 .
:tranSetES-NO/arte-n-SE00007455-sense-kunst-n-arte-n-SE00007455-sense-TC00017354-trans
      a vartrans:Translation ;
     vartrans:source :lexiconES/arte-n-SE00007455-sense;
     vartrans:target :lexiconNO/kunst-n-arte-n-SE00007455-
     sense dc:source :mlds-ES3 .

:lexiconES/arte-n-SE00007455-sense-TC00017355-ex a
  lexicog:UsageExample ; rdf:value "La música, la danza y la pintura son
  formas de arte."@es ; rdf:value "Muziek, dans en schilderen zijn
  vormen van kunst."@nl ; rdf:value "Musikk, dans og maling er
  kunsttyper."@no .
```

Example 1.6: RDF Turtle representation of an extract of the Spanish entry *arte* with translations into Dutch and Norwegian, examples, and translations of the examples.

Each ontolex:LexicalEntry has an ontolex:LexicalSense, which is the bridge between the linguistic description and the semantic layer, following the notion of *semantics by reference* embraced in *lemon*.[12] The example is recorded through an instance of lexicog:UsageExample linked to the senses via lexicog:usageExample. Note that this instance has different values, each for the realization of that example in a different language.

[12] Due to the lack of ontology entities to act as references for ontolex:LexicalSenses, the semantics provided by definitions will be captured through ontolex:LexicalConcepts and the property skos:definition. However, the instantiation of the OntoLex core, beyond *lexicog*, is out of the scope of this paper, and we refer the reader to the examples provided at the *lexicog* documentation page.

6. Conclusions and future work

In this paper we have presented work on applying the new *lexicog* module of OntoLex-*lemon* to KD's multilingual data as a real use case scenario for the extension. We have shown that *lexicog* addresses the gaps previously identified in the literature (Klimek & Brümmer, 2015; Bosque-Gil et al., 2016b, 2017) as regards the loss of structural and implicit lexical information in the original resource, and provides elements to capture data frequently found in lexicographic records, such as usage examples, translations, or annotations on morphosyntatic features. In addition, and to serve as a basis for future transformations of lexicographic data, we framed the modelling with *lexicog* in the whole conversion process of KD to LD. We have detailed the incremental approach followed in the conversion process and outlined the different steps performed as part of the validation process for the resulting RDF.

The next step will be to process the data in a triple store, serving both to further validate the flawless conversion from XML to RDF and to prepare the data for linking to other external LD resources. Then, the actual linking to such external data resources can take place. Future work includes linking between different KD monolingual cores, creating one interconnected, fully cross-lingual graph, as well as linking to external sources, thus enhancing the data and providing even more elaborate and enriched data to Lexicala API users and for various research and development purposes.

The task of linking dictionaries, by associating a translation of a headword in the source language dictionary core to its corresponding entry in the target language dictionary core, is an ambitious and elaborate one. The main hindrance is automating the process, managing to link a translation equivalent to the correct senses across languages, which is ultimately related to word sense disambiguation, and has been previously attempted with KD data as part of the LDL4HELTA project and the Translation Inference Across Dictionaries shared tasks and workshops (TIAD).[13] The conversion of KD monolingual cores to LD has laid the groundwork for this type of graph, and provided further ideas for carrying out this goal in the future.

In the meantime, linking KD data to other sources should be significantly facilitated by the current conversion. Linking KD data with external, annotated or enriched resources, will greatly enhance both its commercial appeal and potential for further research, and can serve as a detailed and efficient resource for language processing and parsing tasks in the realm of computational linguistics, thus expanding the outreach of LD-compliant lexicographic data yet further.

[13] http://2019.ldk-conf.org/tiad-2019/

7. Acknowledgements

This work has been supported by the Spanish Ministry of Education, Culture and Sports through the FPU program, and by the European Union's Horizon 2020 research and innovation programme through the projects Lynx (grant agreement No 780602), Elexis (grant agreement No 731015) and Prêt-à-LLOD (grant agreement No 825182). It has been also partially supported by the Spanish National projects TIN2016-78011-C4-3-R (AEI/ FEDER, UE) and DGA/FEDER.

8. References

Abromeit, F., Chiarcos, C., Fäth, C. & Ionov, M. (2016). Linking the Tower of Babel: Modelling a Massive Set of Etymological Dictionaries as RDF. In *Proceedings of the 5th Workshop on Linked Data in Linguistics: Managing, Building and Using Linked Language Resources (LDL-2016)*. pp. 11–19.

Bizer, C., Heath, T. & Berners-Lee, T. (2009). Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3), pp. 1–22.

Bosque-Gil, J., Gracia, J. & Gómez-Pérez, A. (2016a). Linked data in lexicography. *Kernerman Dictionary News*, (26), pp. 19–24. https://www.kdictionaries.com/kdn/kdn24_2016.pdf.

Bosque-Gil, J., Gracia, J. & Montiel-Ponsoda, E. (2017). Towards a Module for Lexicography in OntoLex. In *Proc. of the LDK workshops: OntoLex, TIAD and Challenges for Wordnets at 1st Language Data and Knowledge conference (LDK 2017), Galway, Ireland*, volume 1899. Galway (Ireland): CEUR-WS, pp. 74–84. http://ceur-ws.org/Vol-1899/OntoLex{_}2017{_}paper{_}5.pdf.

Bosque-Gil, J., Gracia, J., Montiel-Ponsoda, E. & Aguado-de Cea, G. (2016b). Modelling Multilingual Lexicographic Resources for the Web of Data: the K Dictionaries case. In *Proc. of GLOBALEX'16 workshop at LREC'16, Portoroz, Slovenia*.

Declerck, T., Wand-Vogt, E. & Mörth, K. (2015). Towards a Pan European Lexicography by Means of Linked (Open) Data. In I. Kosem, M. Jakubíček, J. Kallas & S. Krek (eds.) *Proceedings of eLex 2015. Biennial Conference on Electronic Lexicography (eLex2015), electronic lexicography in the 21st century: Linking lexical data in the digital age*. Ljubljana/Brighton: Trojina, Institute for Applied Slovene Studies, Ljubljana.

Gracia, J., Villegas, M., Gómez-Pérez, A. & Bel, N. (2018). The apertium bilingual dictionaries on the web of data. *Semantic Web*, 9(2), pp. 231–240.

Kaltenböck, M. & Kernerman, I. (2017). Introducing LDL4HELTA: Linked data lexicography for high-end language technology application. *Kernerman Dictionary News*, (25), pp. 2–3. https://www.kdictionaries.com/kdn/kdn25_2017.pdf.

Kernerman, I. (2009). KD's BLDS: a brief introduction. *Kernerman Dictionary News*, (17), pp. 1–2. http://www.kdictionaries.com/kdn/kdn17_2009.pdf.

Kernerman, I. (2011). From dictionary to database: Creating a global multi-language series. In I. Kosem & K. Kosem (eds.) *Electronic Lexicography in the 21st Century. New Applications for New Users. Proceedings of eLex*, pp. 113–121. http://elex2011.trojina.si/Vsebine/proceedings/eLex2011-14.pdf.

Kernerman, I. (2015). A multilingual trilogy: Developing three multi-language lexicographic datasets. *Dictionaries: Journal of the Dictionary Society of North America*, 36(1), pp. 136–149. http://elex.link/elex2015/proceedings/eLex_2015_24_Kernerman.pdf.

Klimek, B. & Brümmer, M. (2015). Enhancing lexicography with semantic language databases. *Kernerman DICTIONARY News*, 23, pp. 5–10. https://www.kdictionaries.com/kdn/kdn23_2015.pdf.

Krek, S., Kosem, I., McCrae, J. P., Navigli, R., Pedersen, B. S., Tiberius, C. & Wissik, T. (2018). European Lexicographic Infrastructure (ELEXIS). In *The XVIII EURALEX International Congress*. p. 159.

McCrae, J., Aguado-de Cea, G., Buitelaar, P., Cimiano, P., Declerck, T., Gómez-Pérez, A., Gracia, J., Hollink, L., Montiel-Ponsoda, E., Spohr, D. & Wunner, T. (2012). Interchanging lexical resources on the Semantic Web. *Language Resources and Evaluation*, 46, pp. 701–719.

McCrae, J., Bosque-Gil, J., Gracia, J., Buitelaar, P. & Cimiano, P. (2017). The OntoLexLemon Model: Development and Applications. In I. Kosem et al. (eds.) *Electronic lexicography in the 21st century. Proc. of eLex 2017 conference, in Leiden, Netherlands*. Lexical Computing CZ s.r.o., pp. 587–597. https://elex.link/elex2017/wp-content/uploads/2017/09/paper36.pdf.

McCrae, J., Tiberius, C., Khan, F. A., Kernerman, I., Declerck, T., Krek, S. Monachini, M. & Ahmadi, S. (2019). The ELEXIS Interface for Interoperable Lexical Resources. Deliverable, ELEXIS-European Lexicographic Infrastructure.

Parvizi, A., Kohl, M., González, M. & Saur', R. (2016). Towards a Linguistic Ontology with an Emphasis on Reasoning and Knowledge Reuse. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.*