The Challenges of Converting Legacy Lexical Resources to Linked Open Data using Ontolex-Lemon: The Case of the Intermediate Liddell-Scott Lexicon

Fahad Khan, Andrea Bellandi, Federico Boschetti, and Monica Monachini

Istituto di Linguistica Computazionale "A. Zampolli" - CNR,
Pisa, Italy
{fahad.khan,andrea.bellandi,federico.boschetti,monica.monachini}@ilc.cnr.it

Abstract. In this article we discuss the conversion of a legacy lexical resource, an abridged version of the ancient Greek-English lexicon, the Liddell-Scott-Jones lexicon, into RDF using the lemon model discussing some of the challenges we confronted during this conversion. We will also introduce the polyLemon vocabulary which we introduced to describe the structuring of the senses in a lexical entry in a dictionary.

Keywords: lemon-ontolex, Liddell-Scott lexicon, Ancient Greek

1 Introduction

Although the publication of language resources as Linked (Open) Data is being seen as increasingly important within the language resources and technologies community, a look at the LLOD cloud[1] reveals that there is still a lack of lexical resources dealing with 'historical' languages such as ancient Greek, Latin or Sanskrit. This can be seen as a missed opportunity for two reasons. The first is that there already exist numerous legacy print resources dealing with these languages (especially Latin and Greek) and which are now out of copyright. These can be digitised and after some amount of manual curation, converted into the RDF format, and consequently made freely available as Linked Open datasets. The second reason is that in many cases these languages are still being taught in schools and universities and so these resources already have a large ready made audience. In this article we will look at the conversion of a legacy lexical resource, an important 19th century Ancient Greek -English dictionary, the Intermediate Liddell Scott Jones Greek-English lexicon, colloquially known as the Middle Liddell (ML), into RDF using the Ontolex-Lemon model.

Fortunately, the actual difficult work of digitising the original print dictionary source and converting into a usable computational format with most of the

[1] http://linguistic-lod.org/llod-cloud

salient structural information already marked out and annotated, in this case TEI, had already been done for us by the Perseus project[2]. That is, we were able to take advantage of the fact Perseus project had already made the ML available with an open license, along with an abundant amount of other language resources – and not only in Greek and Latin but also in a number of other languages such as Icelandic and Arabic. Indeed in the near future we are planning to convert and publish some of these other Perseus lexica as LOD too, including the full version of the Liddell-Scott-Jones (LSJ) lexicon [1], and the Lewis-Short Latin-English lexicon. We feel that the LSJ especially would make an important addition to the LLOD cloud both because of its historical influence as well as its continuing relevance and use by to students and scholars of the Ancient Greek language. However, due to the complexity of the original resource we decided to begin with the Middle Liddell (ML) [3]. In the course of the conversion of the Perseus TEI-XML encoding of the resource into RDF we came across a number of issues which we think have a wider bearing on the conversion of legacy lexicographic resources into LD and which we feel would make this an interesting case study. In the next section we will focus on one of these, namely the use of the Ontolex-Lemon model in the conversion of the ML into LD.

2 Using Ontolex-Lemon

One of the most important aspects of the publication of datasets in RDF is the use and re-use of models/vocabularies which allow the explicit encoding of pertinent aspects of the dataset to be modelled. Indeed the re-use of models, standards and vocabularies, is one of the core best practices underpinning the linked open data publishing paradigm. This means in effect that anyone who wants to publish data as linked open data is strongly encouraged, in the interests of interoperability, to check for the availability of already existing vocabularies which fulfil the modelling requirements of the dataset in question.

Perhaps the single most popular model for modelling and representation of lexical datasets as RDF is the lemon model (LExicon MOdel for ONtologies) [4]. A second, updated version of the lemon model, Ontolex-Lemon, with the addition of new modules and a significant number of other changes, was published last year. We had originally started off the conversion using the previous version of lemon but then afterwards decided to use the newer version. In what follows we will use 'lemon' to refer to this latest version Ontolex-Lemon unless otherwise specified.

One important factor to take into consideration here is that lemon was originally proposed as a model for helping enrich ontologies with linguistic information and not for converting data arising from already existing lexicographic resources and that conforms to certain conventions of printed dictionaries [4]. For example the lemon model requires that each lexical sense is linked to a reference object that describes the extension of the related lexical entry; this is given

[2] `http://www.perseus.tufts.edu/`

in the form of an OWL axiom. In many print dictionaries however the senses of a single entry may be nested in order to give a more complex description of the meaning of a word, and often it doesnt seem necessary or even viable for each individual sense listed in a dictionary to be linked to its own separate concept. And so it's important to be able to represent this and other structural aspects of the original data. In what follows we will look at the additional classes and properties that we have defined and that fall outside the scope of those already included in the Ontolex-Lemon specifications.

Although we cannot go into much depth in this article on the different approaches to representing print dictionaries using a computational model like RDF or LMF, and in particular how faithful to be in representing different aspects of the organisation of the original resource, we will nevertheless touch on these and related issues in what follows.

3 The Source Dataset: Perseus's TEI-XML Encoding of the Middle Liddell

In the course of carrying out background research on traditional lexicographic resources, we found that the complex nested structure that one sees in the ML was actually very common in other scholarly or comprehensive print dictionaries such as the Lewis-Short Latin-English lexicon and the Oxford English Dictionary. Within the TEI-DICT guidelines this nesting is captured by the use of the `@level` attribute of the sense element. For instance take the entry from the ML given in Fig. 1, where the different levels of nesting are labeled using Roman and Arabic numerals[3]. The convention, in the ML, as well as in the original Liddell-Scott-Jones lexicon and in a number of other similar dictionaries, is to label these levels using both Roman and Arabic numerals as well as capital Roman alphabet letters and small Roman alphabet letters, depending on the level of nesting.

As Fig. 1 show the senses in the ML area effectively organised in a tree structure. The Perseus TEI-DICT XML version of this entry is shown in Fig. 2. When it came to representing this sense tree structure in RDF, and given that however we decided to create an extension of the lemon core model, called polyLemon[4].

PolyLemon consists of the object properties `senseSibling`, `senseChild` and `senseDescendant` and the datatype properties `senseLevel` and `senseID` all of which help to determine the position of a sense in the sense tree of a lexical entry.

Figure 5 represents, in diagrammatic form, the polyLemon based RDF encoding of the sense structure of the ML entry given in Figs. 1 and 2. The horizontal arrows represent instances of the `senseSibling` property and the vertical/slanted lines instances the `senseChild` property.

---

[3] This entry can be accessed via the Perseus Hopper here `http://www.perseus.tufts.edu/hopper/`.

[4] `http:\\lari-datasets.ilc.cnr.it/polyLemon`

ἀληθής α privat. , λήθω ˜ λανθάνω

unconcealed, true:

I._true_, opp. to ψευδής, **Hom.**; τὸ ἀληθές, by crasis τἀληθές, ionic τὠληθές, and τὰ ἀληθῆ, by crasis τἀληθῆ _the truth_, **Hdt.**, attic

2.of persons, _truthful_, **Il.**, attic

3.of oracles and the like, _true, coming true_, **Aesch.**, etc.

II.adv. ἀληθῶς, ionic -θέως, _truly_, **Hdt.**, etc.

2._really, actually, in reality_, **Aesch.**, **Thuc.**, etc.; so, ὡς ἀληθῶς **Eur.**, **Plat.**, etc.

III.neut. as adv., proparox. ἄληθες; _itane? indeed? really? in sooth?_ ironically, **Soph.**, **Eur.**, etc.

2.τὸ ἀληθές _really and truly_, Lat. revera, **Plat.**, etc.; so, τὸ ἀληθέστατον _in very truth_, **Thuc.**

Fig. 1. Example ML Entry.


```
<entry key="a)lhqh/s" type="main" id="n1401">
  <form>
    <orth extent="full" lang="greek">a)lhqh/s</orth>
  </form>
  <etym>
    <foreign lang="greek">a</foreign> privat. <!--a_privat-->, <foreign lang="greek">
   lh/qw &amp;equals; lanqa/nw</foreign>
  </etym>
  <sense level="0" n="0" id="n1401.0"><trans><tr>unconcealed, true</tr></trans>:
   </sense>
  <sense level="2" n="I" id="n1401.1"><trans><tr>true</tr></trans>, opp. to <foreign
   lang="greek">yeudh/s</foreign>, <usg>Hom.</usg>; <foreign lang="greek">to\
   a)lhqe/s</foreign>, by crasis <foreign lang="greek">ta)lhqe/s</foreign>, ionic
   <foreign lang="greek">tw)lhqe/s</foreign>, and <foreign lang="greek">ta\ a)lhqh=
   </foreign>, by crasis <foreign lang="greek">ta)lhqh=</foreign><trans><tr>the truth
   </tr></trans>, <usg>Hdt.</usg>, attic</sense>
  <sense level="3" n="2" id="n1401.2">of persons, <trans><tr>truthful</tr></trans>, <usg>
   Il.</usg>, attic</sense><sense level="3" n="3" id="n1401.3">of oracles and the like,
   <trans><tr>true, coming true</tr></trans>, <usg>Aesch.</usg>, etc.</sense>
  <sense level="2" n="II" id="n1401.4">adv. <ref targOrder="U" lang="greek">a)lhqw=s
   </ref>, ionic <foreign lang="greek">-qe/ws</foreign>, <trans><tr>truly</tr></trans>,
   <usg>Hdt.</usg>, etc.</sense>
....
</entry>
```


Fig. 2. Perseus TEI-XML Encoding.

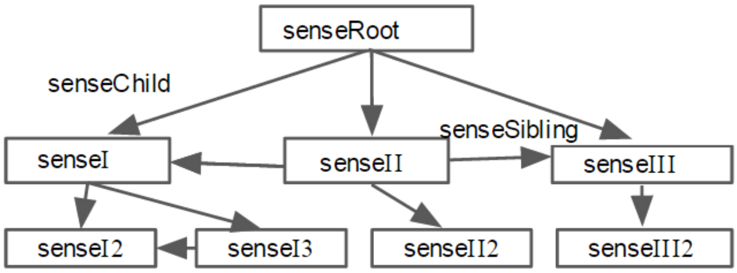

Fig. 3. Sense tree.

It is important to note here that, as Lew argues in [2], the term sense is not always necessarily used in the same way within practical lexicography as it is in other fields of linguistics or in computational linguistics. In the latter case it is used, more often than not, to denote the intensional aspects of word meaning, as distinguished from the extensional components of meaning (the references of a word); whereas in the former case senses are used to mark out distinct component parts of [a] dictionary article, and serve the purpose of assisting the user in whatever lexicographically relevant queries problems and doubts they may have [2]. These two different approaches to defining the notion of sense might do not always necessarily line up with each other. Therefore we might question the use of `lemon:sense` in this instance and even the use of the lemon model at all, given that we are dealing with what is clearly a lexicographic resource where the senses have been arranged primarily to provide for ease of access rather than according to some formal model of word meaning.

However we felt in the end that this was to be take too puritanical an approach and that in the interests of interoperability and the accessibility of the resource given the popularity and widespread use of lemon – we would stick to the `lemon:LexicalSense` class using polyLemon to describe the senses of each entry in order to specify the fact that we were dealing with a specific type of arrangement to be found mostly in dictionaries. Alternatively the possibility was suggested to us of redefining dictionary senses as `skos:Concept` entities and therefore circumventing the use of polyLemon to define a hierarchical sense structure. Note that entities of the type `skos:Concept` are defined as being independent of the terms used to define them[5], but lexicographic senses, as we might call them, even if they differ from other kinds of word senses, still, arguably, serve to describe the use of words when used with a certain meaning within a certain language community and do not directly describe the referent or conceptual content itself – at least not independently of the lexical entry is associated with the sense (the Lexical Concept class in ontolex is a subclass of

[5] `https://www.w3.org/2009/08/skos-reference/skos.html`

`skos:Concept` and so faces the same difficulties as the latter in this regard). On the other hand it is true that senses in ML are often used to group together more specific senses, a situation – something that could be modeled using skos relations `narrower` and `broader` – although, as we mentioned above, in the case of the ML and many other dictionaries this grouping of senses is primarily intended as means of enabling easy access to senses and not as a robust hierarchical conceptualisation of some domain. And so we decided to err on the side of caution and to not make ML dictionary senses `skos:Concept` entities as we felt that this would have introduced an extra layer of interpretation. In addition we felt that as this hierarchical way of arranging word senses was common enough in traditional lexicographic resources a specialised vocabulary like polyLemon was merited in this case.

Having made the decision to use polyLemon to represent the sense structure of each entry we wrote a script to extract this structure from each lexical entry in the Perseus XML encoding. The rest of the conversion was fairly straightforward. Each lexical entry in our lemon encoding of the ML has both a betacode and a unicode written representation, as well as (when it is explicitly stated in the original Perseus XML dataset) information on part of speech. We used the lexinfo vocabulary to represent this POS data, using the property `lexinfo:partOfSpeech` and the lexical categories available in lexinfo. We used the IDs used to identify lexical entries and senses in the original Perseus version in our lemon version.

As an example see Fig 4 the lexical entry for *dolichos* meaning 'race'[6]. Luckily,

```
:lsjEntry_n8720 a ontolex:LexicalEntry ;
   lsj-lemon:lsjID "n8720" ;
   lexinfo:partOfSpeech lexinfo:Noun ;
   ontolex:canonicalForm :lsjEntry_n8720canForm ;
   ontolex:sense :lsjsense_n8720_0 .
```

Fig. 4. The lexical entry for *dolichos*.

most of the information that we wanted to include for each Lexical Sense object in the RDF encoding of the ML, was already marked up within text contents of the sense elements in the Perseus TEI-DICT version of the ML. For instance we can extract a gloss from the text content of each sense element. In our RDF encoding we provide this gloss as a string stripped of all the XML tags present in the original using an adhoc property `strippedForm` that we have defined for this purpose. This enables users of our resource to peruse the text of the original entry.

---

[6] Note that the lsj-lemon namespace contains a number of classes and properties which we considered to be useful both for the encoding the Middle Liddell and the original LSJ and which we therefore put into a separate file.

One of the elements that is marked up within the text content of sense elements in the Perseus XML source is the translation of a sense. This is marked up using `trans` and `tr` elements. We therefore extracted the translation as a string and linked it to the relevant `lemon:LexicalSense` object used the `lexinfo:translation property`. In future we plan to link each sense to an appropriate Wordnet synset.

We made a decision at the start to work on the Middle Liddell as opposed to the full unabridged version of the text, in large part because this made it easier to manually check the resulting conversion of the dictionary. However Perseus have also made a TEI-DICT XML version of the full Liddell Scott Jones dictionary available and provided CTS-URNs for the citations given for each sense; we are also planning to convert this full version in the future. And although this kind of information wasnt encoded in the ML [7], what was included was the name of an author or a corpus in which the sense in question could be located; this is included between the `usg` tags in the Perseus ML. In most cases it was fairly easy to map between the content of these `usg` elements since in many cases this information was included either in the printed version of the ML or on the Perseus website – in others we weren't able to find a relevant DBpedia link. In the current version we have manually linked word senses to DBpedia resource based on what is contained in the `usg` tags, for instance Aesch. would be mapped to `http://dbpedia.org/resource/Aeschylus`.

We end this section with an example of the representation of the sense. We take one of the senses of the word *ephiemi*. The dataset can be accesssed directly at `http://lari-datasets.ilc.cnr.it/lsj`. We are currently developing a SPARQL interface and setting up a pubby interface. A first version of both can be found at `http://lari-lsj.ilc.cnr.it/LSJSPARQLinterface` and `http://lari-lsj.ilc.cnr.it/page`, respectively.

4 Future Work

As mentioned above we plan to convert the whole of the Liddell-Scott-Jones lexicon along with the Lewis-Short Latin-English lexicon. We are also working on improving our interfaces in order to make the lexicon accessible to as many researchers as possible, including humanists as well as more technically skilled users.

References

1. Crane, Gregory, Bamman, David, and Babeu, Alison. "Philology in an Electronic Age." Preprint of a book chapter in Thompson and Fraser "Greek Lexicography after Liddell and Scott".`http://hdl.handle.net/10427/42688`, 2007.
2. Lew, Robert. "Identifying, ordering and defining senses." The Bloomsbury companion to lexicography (2013): 284-302.

[7] The printed version of the ML doesn't give precise citations for textual attestations.

```
:lsjsense_n14292_6 a ontolex:LexicalSense ;
   lsj-lemon:glossWords "concedere"@la ;
   lsj-lemon:lsjlevel "2" ;
   lsj-lemon:lsjn "II" ;
   lsj-lemon:strippedForm """to let go, loosen, esp. the rein, Plat.:
      &mdash; hence to give up, yield, Lat. concedere, τινὶ τὴν ἡγεμονίαν Thuc.:
      &mdash;c. inf. to permit, allow, τινὶ ποιεῖν τι Hdt., Soph., etc.
"""@en ;
   lsj-lemon:usage <http://dbpedia.org/resource/Plato>,
      <http://dbpedia.org/resource/Sophocles>,
      <http://dbpedia.org/resource/Thucydides> ;
   lsj-lemon:usageText "Hdt.",
      "Plat.",
      "Soph.",
      "Thuc." ;
   polyLemon:senseSibling :lsjsense_n14292_9 ;
   lexinfo:translation "to give up, yield"@en,
      "to let go, loosen"@en,
      "to permit, allow"@en ;
   ontolex:isSenseOf :lsjEntry_n14292 .
```

Fig. 5. Sense of the word *ephiemi*.

3. Liddell, Henry George, and Robert Scott. An intermediate Greek-English lexicon: founded upon the seventh edition of Liddell and Scott's Greek-English lexicon. Harper & Brothers, 1896.
4. McCrae, John, Dennis Spohr, and Philipp Cimiano. "Linking lexical resources and ontologies on the semantic web with lemon." Extended Semantic Web Conference. Springer Berlin Heidelberg, 2011.