Language Resources and Linked Data: A Practical Perspective

Jorge Gracia[1(✉)], Daniel Vila-Suero[1], John P. McCrae[2], Tiziano Flati[3], Ciro Baron[4], and Milan Dojchinovski[5]

[1] Ontology Engineering Group, Universidad Politécnica de Madrid, Madrid, Spain
{jgracia,dvila}@upm.es
[2] CITEC, University of Bielefeld, Bielefeld, Germany
jmccrae@cit-ec.uni-bielefeld.de
[3] LCL, Sapienza Università di Roma, Roma, Italy
flati@di.uniroma1.it
[4] AKSW, University of Leipzig, Leipzig, Germany
cbaron@informatik.uni-leipzig.de
[5] Czech Technical University in Prague, Praha, Czech Republic
milan.dojchinovski@fit.cvut.cz

Abstract. Recently, experts and practitioners in language resources have started recognizing the benefits of the linked data (LD) paradigm for the representation and exploitation of linguistic data on the Web. The adoption of the LD principles is leading to an emerging ecosystem of multilingual open resources that conform to the Linguistic Linked Open Data Cloud, in which datasets of linguistic data are interconnected and represented following common vocabularies, which facilitates linguistic information discovery, integration and access. In order to contribute to this initiative, this paper summarizes several key aspects of the representation of linguistic information as linked data from a practical perspective. The main goal of this document is to provide the basic ideas and tools for migrating language resources (lexicons, corpora, etc.) as LD on the Web and to develop some useful NLP tasks with them (e.g., word sense disambiguation). Such material was the basis of a tutorial imparted at the EKAW'14 conference, which is also reported in the paper.

Keywords: Linked data · Language resources · Multilingual web of data

1 Introduction

Linked data (LD) is a set of best practices for exposing, sharing, and connecting data on the Web [2]. Recently, researchers working on linguistic resources have shown increasing interest in publishing their data as LD [4]. Nowadays, there are many good examples involving important organizations and initiatives that stress the opportunities offered by LD and foster the aggregation of multilingual open resources into the Linked Open Data (LOD) cloud. By interlinking multilingual

P. Lambrix et al. (Eds.): EKAW 2014 Satellite Events, LNAI 8982, pp. 3–17, 2015.
DOI: 10.1007/978-3-319-17966-7_1

and open language resources, the Linguistic Linked Open Data (LLOD) cloud is emerging[1], that is, a new linguistic ecosystem based on the LD principles that will allow the open exploitation of such data at global scale. In particular, these are some key benefits of linguistic LD:

- Provide enhanced and more sophisticated navigation through multilingual data sets and linguistic data
- Support easier integration of linguistic information into research documents and other digital objects
- Support easier integration of linguistic information with LOD datasets, enhancing the natural language description of those datasets
- Facilitate re-use across linguistic datasets, thus enriching the description of data elements with information coming from outside the organization's local domain of expertise
- Describe language resources in RDF [10] and make them indexable by semantic search engines
- Avoid tying developers and vendors to domain-specific data formats and dedicated APIs.

With the aim of contributing to the development of the LLOD cloud, we organised a tutorial at the EKAW'14 conference[2] on the topic "Language Resources and Linked Data". The tutorial tackled the following questions:

1. How to represent rich multilingual lexical information (beyond `rdfs:label`) and associate it to ontologies and LD?
2. How to generate multilingual LD from data silos?
3. How to represent multilingual texts, annotations and corpora as LD?
4. How to perform word sense disambiguation and entity linking of LD?

The tutorial aimed at answering the above questions in a practical way, by means of examples and hands-on exercises. In this paper, we summarise different theoretical and practical aspects concerning the representation and publication of LLOD on the Web, and give a summary of the mentioned tutorial including pointers to the educational material and practical exercises used on it.

The remainder of the paper is organised as follows. In Section 2, the patterns based on the *lemon* model for representing ontology lexica are introduced. Section 3 summarises a methodology for generating and publishing multilingual linguistic LD. In Section 4, we show how to integrate NLP with LD and RDF based on the NIF format. In Section 5, disambiguation and entity linking methods based on BabelNet are explained. Section 6 summarises the outline and outcomes of the EKAW'14 tutorial on "Language Resources and Linked Data" and, finally, conclusions can be found in Section 7.

[1] A picture of the current LLOD cloud can be found at http://linghub.lider-project.eu/llod-cloud. The picture was jointly developed by the Open Knowledge Foundation's Working Group on Open Data in Linguistics (http://linguistics.okfn.org) and the LIDER project (http://www.lider-project.eu/).

[2] http://www.ida.liu.se/conferences/EKAW14/home.html

2 Modelling Lexical Resources on the Web of Data: The *lemon* Model

In this section we will see how to represent rich lexical information associated to ontologies and LD, and how to use a set of design patterns to facilitate such representation in a practical way.

2.1 Modelling Ontology-Lexica

Lexical resources such as WordNet [14] are one of the most important types of data sources for linguistic research. Such resources are complementary to another type of Web resources that contain a large amount of taxonomic data described in RDF such as DBpedia [3]. Bridging the gap between these two types of resources means that rich linguistic information found in lexical resources (e.g., lexicons) can be used to describe information on the Web, enabling novel applications such as question answering over LD [18]. This leads to a new type of resources that is termed *ontology-lexicon*, which consist of an ontology describing the semantic and taxonomic nature of the domain and a lexicon describing the behaviour of the words in a language.

Building on the existing work of models such as LexInfo [5] and LIR [15], *lemon* (Lexicon Model for Ontologies) [11] was proposed to provide a "de facto" standard which is used by a cloud of lexical LD resources such as WordNet [13], BabelNet [7], and UBY [6] among many others. The *lemon* model's core consists of the following elements depicted in Figure 1:

Lexical Entry. A lexical entry, which may be a word, multiword expression or even affix, is assumed to represent a single lexical unit with common properties, especially part-of-speech, across all its forms and meanings.

Lexical Form. A form represents a particular version of a lexical entry, for example a plural or some other inflected form. A form may have a number of representations in different orthographies (e.g., spelling variants) or media (e.g., phonetic representations).

Lexical Sense. The sense refers to the usage of a word with a specific meaning and can also be considered as a reification of the pair of a lexical entry used with reference to a given ontology. The sense is also used as a node for the annotation of many pragmatic features such as register.

Reference. The reference is an entity in the ontology that the entry can be interpreted as, or alternatively that can be represented by using the lexical entry.

In addition to the core, *lemon* provides a number of models to enable representation and application of the model to a wide variety of domains. Firstly the **linguistic description** module enables annotations to be added to entries, forms or senses. Secondly, the **phrase structure** module allows description of how the words within a multiword entry relate. Next, the **syntax and mapping** module is used to represent how a syntactic frame corresponds to one or more

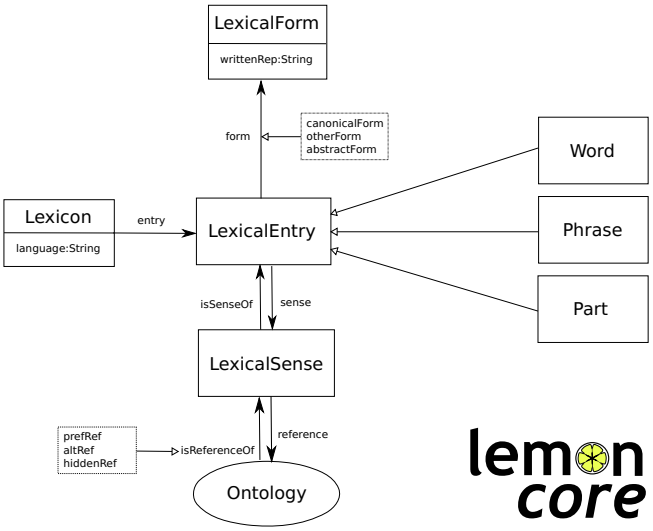

Fig. 1. The *lemon* core model, indicating the classes, the properties used to connect them and their subclasses.

semantic predicates in the ontology. The **variation** module captures the representation of variants between terms. Finally, the **morphology** module allows for regular expression representation of morphological variants avoiding the need to represent many forms for a single lexical entry.

Additional extensions have been also developed, such as the **translation** module [8], which allows for representing explicit translations between lexical senses documented in different natural languages, or the **lemon-BabelNet** extension required for the LD representation of BabelNet (see Section 5).

2.2 The *Lemon* Design Pattern Language

When creating a lexicon from scratch, common patterns quickly emerge for the creation of lexical entries and associated ontology axioms. These patterns have been assembled into the *lemon* Design Pattern Language[3], which provides a compiler to generate a standard RDF/XML representation of the data.

These patterns describe the ontological type and the part-of-speech of the entry, such as for example the 'class noun' pattern which describes a noun referring to a class in the ontology. In addition, to the triples stating these two facts the entry is also associated with a noun predicate frame ('X is a N'). The 'object property noun' pattern is similar but takes as parameters not only the lemma but also a property (p) and an individual (v) and generates an anonymous class (C) with the axiom $C \equiv \exists p.v$ which is associated with the noun like in the 'class

[3] http://github.com/jmccrae/lemon.patterns

noun' pattern. Examples of these patterns are given below, which indicates that "cat" refers to a class `dbr:Cat` the noun "German" refers to all elements whose `dbp:nationality` has a value of `dbr:Germany`:

```
@prefix dbr: <http://dbpedia.org/resource/> .
@prefix dbp: <http://dbpedia.org/property/> .

Lexicon(<http://www.example.com/lexicon>,"en",
  ClassNoun("cat", dbr:Cat),
  ObjectPropertyNoun("German", dbp:nationality, dbr:Germany))
```

Verbs are divided into state verbs which express a general ontological fact and consequence verbs where the ontological fact is a consequence of the event described by the verb[4]. In addition there are patterns to model verbs with more than two arguments to multiple ontology predicates. For adjectives, there are also patterns describing adjectives as classes ('intersective adjectives'), comparable properties ('scalar adjectives') and relative adjectives, which is described more completely in McCrae et al. [12].

3 Methodology for Multilingual Linguistic Linked Data Generation

The previous section presented a way of representing lexical data in RDF. However, representing linguistic data is only a part of the whole process required to expose language resources as LD. In this section we will give an overview of such process and some methodological guidelines.

In fact, several guidelines [20] have been proposed to produce and publish LD on the Web. These guidelines are meant to provide a set of tasks and best practices to generate and make available high quality LD. More recently, Vila et al. [19] proposed general guidelines for generating *multilingual* LD. In addition, the W3C Best Practices for Multilingual Linked Open Data community group[5] has recently published specific guidelines for generating and publishing LD out of several types of language resources (e.g., bilingual dictionaries, WordNets, terminologies in TBX, etc). The core activities identified in such guidelines are[6]: (i) selection of vocabularies, (ii) RDF generation, and (iii) publication.

Selection of vocabularies. In this activity the goal is to select standard vocabularies to represent linguistic data. The diverse options depend on the type of data. For example the *lemon* model, described in Section 2, is an appropriate vocabulary for representing lexica, and NIF, described in Section 4, to represent annotations in text. For other (non linguistic) information associated to the language resource, other extendedly used vocabularies can be used such as Dublin Core[7] for provenance and authoring or DCAT[8] for metadata of the RDF dataset.

[4] Generally, state verbs express the triple in the present tense, e.g., 'X knows Y', and consequence verbs express the triple in the past tense, e.g., 'X married Y'

[5] http://www.w3.org/community/bpmlod/

[6] See for instance http://bpmlod.github.io/report/bilingual-dictionaries

[7] http://purl.org/dc/elements/1.1/

[8] http://www.w3.org/ns/dcat#

Furthermore, if we need to model more specific features, a growing number of vocabularies is available on the Web and they can be found in catalogs such as the Linked Open Vocabularies catalog (LOV)[9].

RDF generation. After selecting the vocabularies that will be used to model the linguistic data sources, the main steps to be performed are: (a) modelling the data sources, (b) design the identifiers (URIs) for the RDF resources that will be generated, and (c) transform the data sources into RDF by mapping them to the selected vocabularies and following the identifier patterns previously designed.

Publication. The last activity can be organized in two tasks: (a) dataset publication, and (b) metadata publication. As for dataset publication, there are several architectures available such as having a triple or quad-store to persist and query the data and setting up what is known as LD front-ends, which are basically an access layer on top of a triple-store.

As an example of the above steps, let us imagine that we want to transform a set of spreadsheets containing a set of terms in one language and their correspondent translations into another language. The first step would be to select an appropriate vocabulary and in this case *lemon* and its translation module[10] are a good choice. The next step would be to decide how to model the data contained in the spreadsheets. For instance, we could decide to create a separate lexicon for each language, each row in the spreadsheet corresponding to a different `lemon:LexicalEntry`, and interlink them through translation relations. The following step would be to define the identifier scheme, that is, how URIs are created. There exist several guidelines to design URIs such as the one published by Interoperability Solutions for European Public Administrations [1]. For instance, if the files contain unique identifiers for each lexical entry we could use those identifiers to create the URIs of the lemon lexical entries and append them to a namespace that we own and where we will publish the data. Finally, the last step would be to map and transform the sources into RDF. There are various open source tools to generate RDF depending on the type and format of data[11]. In our case, LODrefine[12] provides an easy way to transform many kinds of tabular and hierarchical data into RDF and its interface is similar to commercial tools to work with spreadsheets. The result of this transformation is the set of RDF files that have to be published on the Web. Finally, regarding publication, LODrefine provides an automatic way to upload the data into Virtuoso, which is a triple store available as open source[13], or we could manually load the data into light-weight alternatives such as Fuseki[14].

[9] http://lov.okfn.org
[10] http://purl.org/net/translation
[11] See for example http://www.w3.org/2001/sw/wiki/Tools
[12] http://sourceforge.net/projects/lodrefine/
[13] https://github.com/openlink/virtuoso-opensource
[14] http://jena.apache.org/documentation/serving_data/

4 Integrating NLP with Linked Data: The NIF format

In the above sections we have explored how to represent lexical information in *lemon* and how to generate and publish it as LD on Web. However, in addition to lexica, the representation and publishing as LD of multilingual texts, annotations and corpora is also important. In this section we will explore the use of the *NLP Interchange Format* (NIF) to that end.

NIF [9] is an RDF/OWL based format which provides all required means for the development of interoperable NLP services, LD enabled language resources and annotations. Other than more centralized solutions such as UIMA[15] and GATE[16], NIF enables the creation of heterogeneous, distributed and loosely coupled NLP applications. The NIF format is based on a URI scheme for minting URIs for arbitrary strings and content in Web documents. It is supported by the NIF Core Ontology[17] which formally defines classes and properties for describing substrings, texts, documents and the relations among them.

The following code presents an example of a simple NIF document with an annotated substring. We will further use this example to explain the NIF basics.

```
1  @base <http://example.com/exampledoc.html#> .
2  <char=0,> a nif:Context , nif:RFC5147String ;
3  <char=86,90>
4          a   nif:RFC5147String , nif:String , nif:Word ;
5          nif:beginIndex          "86"^^xsd:nonNegativeInteger ;
6          nif:endIndex            "90"^^xsd:nonNegativeInteger ;
7          nif:isString            "July" ;
8          nif:referenceContext    <char=0,> ;
9          itsrdf:taIdentRef       dbpedia:July .
10         nif:oliaLink penn:NN .
11         nif:oliaCategory olia:Noun .
```

NIF Basics. Every document in NIF is represented using the `nif:Context` concept and identified using a unique URI identifier (line 2). Further, each annotated substring is represented using the `nif:String` concept, or more specifically, as `nif:Word` (line 4) or `nif:Phrase` concepts. The substrings are also uniquely identified with URI identifiers (line 3). The surface forms of the substrings and document's content are referenced as literals using the `nif:isString` property (line 7). Each substring, using the `nif:referenceContext` property is linked with the corresponding document where it occurs (line 8); an instance of the `nif:Context` class. The begin and end indices are also attached to each substring and document using `nif:beginIndex` and `nif:endIndex` (lines 5–6).

NIF is also aligned with well–established linguistic ontologies and standards such as the Ontologies of Linguistic Annotation (OLiA) and Internationalization Tag Set 2.0 (ITS). OLiA provides NLP tag sets for morpho-syntactical annotations. In NIF it can be used, for example, to assign grammatical category to a `nif:Word` (lines 10–11). ITS 2.0 standardizes attributes for annotating XML and HTML documents with processing information, however, it

[15] https://uima.apache.org/
[16] https://gate.ac.uk/
[17] http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core/nif-core.html

also provides an ontology, which can be reused in NIF. For example, using the `itsrdf:taIdentRef` property we can link particular substring representing a named entity mention with its corresponding DBpedia resource (line 9).

Annotating Strings with NIF. Strings can be easily annotated with NIF using Command Line Interface (NIF–CLI) or using Web service (NIF–WS) implementations. Both methods share a similar set of parameters. This includes, for example, parameters for specifying the input and output format, the base prefix URI for the newly minted URIs and the input text submitted for processing. In the following example we show the annotation of a string using NIF–CLI implementation for the Snowball Stemmer[18]. The result for the submitted text will be a single `nif:Context` document, all the `nif:Word`(s) present in the text and the stem for each word (line 9).

```
java -jar snowball.jar -f text -i 'My favorite actress is Natalie Portman.'
```

```
1  @base <http://example.com/exampledoc.html#> .
2  <char=0,> a nif:Context , nif:RFC5147String ;
3  <char=3,11>
4          a   nif:RFC5147String , nif:Word ;
5          nif:isString          "favorite" ;
6          nif:beginIndex        "3"^^xsd:nonNegativeInteger ;
7          nif:endIndex          "11"^^xsd:nonNegativeInteger ;
8          nif:referenceContext  <char=0,> ;
9          nif:stem              "favorit" ;
```

Using NIF-WS we can expose a particular NLP functionality of a tool (e.g. tokenization, POS tagging or Named Entity Recognition (NER)) on the Web. Hence, it is not necessary to download, setup and run the NIF software making possible the creation of a small NIF corpus using an available NIF-WS service. Some of the implementations which are already exposed as Web services includes Stanford NLP[19], DBpedia Spotlight, Entityclassifier.eu[20], Snowball Stemmer and OpenNLP[21]. The following URL exemplifies the annotation of the string "I'm connected." using the Snowball Stemmer NIF-WS implementation.

```
http://snowball.nlp2rdf.aksw.org/snowball?f=text&i=I'm+connected.&t=direct
```

The parameters used for NIF-WS are similar to NIF-CLI implementations and are summarized at the API specification website[22]. For instance, *informat* (f) specifies the input format, *input* (i) holds the actual string that will be annotated, and *intype* (t) defines how the input is accessed (directly from stdin, from an URL or file).

Querying NIF Annotations. Existing NIF corpora or created RDF documents with NIF annotated strings can be further queried, for example, using a SPARQL interface. Twinkle[23] is a simple tool for loading RDF corpora and querying it using standard SPARQL. Upon starting the Twinkle tool (`java -jar`

[18] http://snowball.tartarus.org/
[19] http://nlp.stanford.edu/software/
[20] http://entityclassifier.eu/
[21] https://opennlp.apache.org/
[22] http://persistence.uni-leipzig.org/nlp2rdf/specification/api.html
[23] http://www.ldodds.com/projects/twinkle/

`twinkle.jar`), we can load the corpora such as the Brown corpus (`File` button), write a SPARQL query (e.g. list all words in a document) and execute it (`Run` button).

```
prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix olia: <http://purl.org/olia/brown.owl#>

SELECT  ?uri, ?word WHERE {
        ?uri a nif:Word.
        ?uri nif:anchorOf ?word
}
```

The query in the example above will return all the words along with their URI identifiers.

```
<char=4405,4407> "he"^^<http://www.w3.org/2001/XMLSchema#string>
<char=7596,7599> "had"^^<http://www.w3.org/2001/XMLSchema#string>
<char=2031,2034> "set"^^<http://www.w3.org/2001/XMLSchema#string>
<char=9916,9922> "reform"^^<http://www.w3.org/2001/XMLSchema#string>
...
```

When querying a document that contains POS tags it is possible to create elaborated queries, for example selecting nouns, verbs and OLiA links using the OLiA mapping.

Available NIF Corpora. A number of NIF corpora from different domains and sizes have been published in the NIF format. For instance, the *N3 collection*[24] of datasets, which can be used for training and evaluation of NER systems. *Wikilinks*[25] is a very large scale coreference resolution corpus with over 40 million mentions of over 3 million entities. It is available in the NIF format and published following the LD Principles. The Brown corpus[26] is another showcase corpus of POS tags in NIF. The NIF dashboard[27] contains a list of these corpora as well as their access address and size. Adding a new corpus to the list is possible by uploading a description file using the DataID[28] ontology.

NIF Resources and Software. The *NIF dashboard* exposes the current status of NIF Web services, as well as access URL, demos, converted corpora, wikis and documentation. The NLP2RDF website[29] contains the last NIF related news and resources of previous publications.

NIF Combinator is a Web application which allows to combine output from multiple NIF-WS in a single RDF model. It is possible, for example, to annotate a string using Stanford NLP and then perform NER using the DBpedia Spotlight Web service, this way creating an enriched corpora. When a corpus is created it is recommended to validate it. For this task another useful application is the *NIF validator* which uses the framework RDFUnit and grants the validation of NIF annotated documents.

[24] http://aksw.org/Projects/N3NERNEDNIF.html
[25] http://wiki-link.nlp2rdf.org/
[26] http://brown.nlp2rdf.org/
[27] http://dashboard.nlp2rdf.aksw.org
[28] http://wiki.dbpedia.org/coop/DataIDUnit
[29] http://nlp2rdf.org/

The source code of the NIF related software is available at *NLP2RDF GitHub web page*[30]. The NIF core engine is developed in Java and the RDF models are manipulated using Apache Jena[31]. The NIF reference code is flexible and implementations for new NLP tools might be done extending NIF classes. The NIF packages also provide helpers for tokenization and creation NIF-CLI and NIF-WS interfaces. In addition, the GitHub repository is used to maintain the core NIF ontology.

5 Multilingual WSD and Entity Linking on the Web

In the previous sections we have focused on how to represent, publish and make linguistic data interoperable on the Web. In the following paragraphs we will review some useful NLP-related tasks that can be done with such data. In fact, the recent upsurge in the amount of information published on the Web requires search engines and machines to analyze and understand text at sense level and in any language. News aggregators and user recommendation systems, for instance, often have the problem to suggest new information to the user such as places or celebrities. For example, in the following sentence it would be useful to understand the senses intended for `Mario` and `strikers`.

```
Thomas and Mario are strikers playing in Munich.
```

This task, however, is affected by the lexical ambiguity of language, an issue addressed by two key tasks: Multilingual Word Sense Disambiguation (WSD), aimed at assigning meanings to word occurrences within text, and Entity Linking (EL), a recent task focused on finding mentions of entities within text and linking them to a knowledge base. The goal shared by the two task is to have multilingual information disambiguated/linked so as to perform better text understanding.

On the one hand EL systems have always been concerned with identifying and disambiguating mentions of named entities only (e.g., `Thomas`, `Mario` and `Munich` are three valid mentions), while WSD algorithms are supposed to disambiguate open class words such as nouns, verbs, adjectives and adverbs (e.g., `strikers` and `playing` are two target words needing disambiguation). The main difference between WSD and EL is thus in the inventory used: the former draws word senses from dictionaries, which usually encode only open and close class words, the latter are instead supposed to link mentions of named entities to concepts to be found in encyclopaedias, such as Wikipedia, DBpedia, etc.

Babelfy [16] is a state-of-the-art WSD/EL system which for the first time solves the two problems jointly, by using BabelNet [17] as the common sense inventory, both for WSD and EL.

5.1 BabelNet

BabelNet is a huge multilingual semantic network at the core of which is the integration of the encyclopaedic information coming from Wikipedia and the

[30] https://github.com/NLP2RDF/
[31] https://jena.apache.org/

lexicographic information of WordNet. By the seamless integration of these resources, BabelNet merges the two sense inventories used by WSD and EL systems separately. With new versions being released (the latest version 3.0 is available at http://babelnet.org) BabelNet now contains more than 13 millions of concepts and named entities lexicalized in 271 languages and has integrated also other several resources such as OmegaWiki, Open Multilingual WordNet, Wiktionary and Wikidata. In order to foster interoperability across linguistic datasets and resources and to further support NLP applications based on the LLD cloud, BabelNet has also been converted into LD [7] by using *lemon* as the reference model (see Section 2) and is also accessible through a SPARQL endpoint.[32] Lemon-BabelNet features almost 2 billion triples and is interlinked with several other datasets including DBpedia as nucleus of the LOD cloud.

By means of the SPARQL endpoint it is possible, for instance, to query the service for all the senses of a given lemma (e.g., *home*) in any language:

```
SELECT DISTINCT ?sense ?synset WHERE {
    ?entries a lemon:LexicalEntry .
    ?entries lemon:sense ?sense .
    ?sense lemon:reference ?synset .
    ?entries rdfs:label ?term .
    FILTER (str(?term)="home")
} LIMIT 10
```

or to retrieve definitions for a given concept (e.g., the first sense of *home* in English) in any language:

```
SELECT DISTINCT ?language ?gloss WHERE {
    <http://babelnet.org/rdf/s00000356n> a skos:Concept .
    OPTIONAL {
        <http://babelnet.org/rdf/s00000356n> bn-lemon:definition ?definition .
        ?definition lemon:language ?language .
        ?definition bn-lemon:gloss ?gloss .
    }
}
```

5.2 Babelfy

Babelfy[33] is a unified, multilingual, graph-based approach to EL and WSD which relies on BabelNet as the background knowledge base from which to draw concepts and lexicalizations to perform the identification of candidate meanings.

From the task point of view, EL on the one hand involves first recognizing mentions contained in text (fragments of text representing named entities) and then linking the mentions to some predefined knowledge base, on the other hand WSD has mentions already identified and consists in selecting the right sense for the word of interest. By generalizing the idea of mention, instead, be it either a named entity or a concept, Babelfy unifies the two sides of the coin by tackling two problems in one. The joint disambiguation and entity linking is performed in three steps:

[32] http://babelnet.org/sparql/
[33] http://babelfy.org/

- Each vertex of the BabelNet semantic network, i.e., either concept or named entity, is associated with a semantic signature, that is, a set of related vertices. This is a preliminary step which needs to be performed only once, independently of the input text.
- Given an input text, all the linkable fragments are extracted from this text and, for each of them, the possible meanings are listed, according to the semantic network.
- The candidate meanings of the extracted fragments are interlinked using the previously-computed semantic signatures, so as to obtain a graph-based semantic interpretation of the whole text. As a result of the application of a novel densest subgraph heuristic high-coherence semantic interpretations for each fragment are finally selected.

A possible application of Babelfy is to easily disambiguate and produce multilingual LD starting from free text written in any language, such as snippets returned by search engines or recommendation websites. The free text is initially enriched with semantic links thanks to Babelfy and then transformed into LD by using the NIF model (cf. Section 4). This feature is of particular interest to the LD community since it provides a means for true interoperability across sense-tagged datasets, one of the concepts at the foundations of the NIF format. In order to produce LD from free text with Babelfy, the following steps are needed:[34]

1. Open the configuration file *babelfy2nif.properties* under the *babelfy/config/* directory;
2. Set up the appropriate parameters so as to account for the language of interest, as well as for the output format (turtle, n-triples or rdf/xml) and type of stream (file vs. standard output). It is also possible to customize the conversion by choosing the algorithm for handling overlapping annotations (either `LONGEST-ANNOTATION-GREEDY-ALGORITHM` or `FIRST-COME-FIRST--SERVED ALGORITHM`). Since Babelfy by default enriches the text with all the possible semantic annotations (including annotations of short, long and even overlapping fragments) these algorithms allow to discriminate annotations whose fragments overlap: the former selects the annotations from the longest to the shortest one, the latter accepts non-overlapping annotations in order of appearance, from the left to the right;
3. Execute the following command '`sh run_babelfy2nif-demo.sh`' (on Linux) or '`./run_babelfy2nif-demo.bat`' (on Windows).

For example, given the sentence "The Semantic Web is a collaborative movement led by the international standards body World Wide Web Consortium (W3C)", an excerpt of the enriched and converted file in NIF is:

[34] All the material referred here (configuration files, executables, etc.) was distributed at the EKAW tutorial and it is now available online (see section 6 for more details).

```
<http://lcl.uniroma1.it/babelfy2nif#char=0,16>
        a                       nif:Word , nif:RFC5147String ;
        nif:anchorOf            "The Semantic Web" ;
        nif:beginIndex          "0" ;
        nif:endIndex            "16" ;
        nif:nextWord            <http://lcl.uniroma1.it/babelfy2nif#char=17,19> ;
        nif:oliaCategory        olia:Noun , olia:CommonNoun ;
        nif:oliaLink            <http://purl.org/olia/penn.owl#NN> ;
        nif:referenceContext    <http://lcl.uniroma1.it/babelfy2nif#char=0,128> ;
        itsrdf:taIdentRef       <http://babelnet.org/rdf/s02276858n> .
```

where the fragment "The Semantic Web" has correctly been linked to the BabelNet synset `<http://babelnet.org/rdf/s02276858n>`.

6 Tutorial on Language Resources and Linked Data

As referred in Section 1, a hands-on tutorial about the above topics was organized on 25th November 2014 in Linköping, Sweden, collocated with the 19th International Conference on Knowledge Engineering and Knowledge Management (EKAW'14) and with the title "Language Resources and Linked Data". The aim of the tutorial was to guide participants in the process of LD generation of language resources in a practical way. This tutorial was the last of a series of related tutorials that were imparted at the International Conference on Language Resources and Evaluation (LREC'14)[35] and at the International Semantic Web Conference (ISWC'14)[36] respectively. The first one was about "Linked Data for Language Technologies"[37] and took place in Reykjavik, Iceland on 26th May. The second took place in Riva del Garda, Italy, on 20th October with the title "Building the Multilingual Web of Data: a Hands-on Tutorial"[38].

The tutorial at the EKAW'14 conference was a full day tutorial divided in five sections: one introductory section and the other four sections covering each of the topics treated previously in this paper (Sections 2 to 5). Each section was divided into a theoretical introduction and a practical session. The practical work consisted of completing some short guided examples proposed by the speakers in order to immediately apply and understand the theoretical concepts. All the instructional material and presentations used in the tutorial were available online in the tutorial's webpage[39] beforehand. Further, a USB pendrive containing all the data and software required to follow the sessions was distributed to every participant. Such material is now available in the tutorial's webpage.

There were no major prerequisites for the attendants to follow the session. Only a certain familiarity with the basic notions of RDF and OWL. Neither previous experience on LD publication nor prior knowledge on NLP techniques or computational linguistics were required. The audience profile ranged from PhD

[35] http://www.lrec-conf.org/lrec2014
[36] http://iswc2014.semanticweb.org/
[37] http://lrec2014.lrec-conf.org/media/filer_public/2013/12/23/t10-tutorialoutline.doc
[38] http://www.lider-project.eu/iswc14_MLWDTutorial
[39] http://www.lider-project.eu/ekaw14_LRLDTutorial

students to research group leaders. Most of them worked in the intersection of Semantic Web and NLP and were interested in exploring the potentiality of linguistic LD in their own research lines. The tutorial was actively followed by 18 attendants (including speakers)[40]. In general, according to feedback received in the discussion session that followed the tutorial, the audience found the experience satisfactory and recognized the interest of an event that covered most of the relevant aspects in the conversion of language resources into LD.

7 Conclusions

In this paper we have summarised different theoretical and practical aspects concerning the representation and publication of LLOD on the Web. In particular we have reviewed: (i) how to represent ontology lexica based on the lemon model, (ii) how to follow a methodology for generating and publishing multilingual linguistic LD, (iii) how to integrate NLP with LD and RDF based on the NIF format and, (iv) how to perform word sense disambiguation and entity linking based on BabelNet. The key notions of such topics have been presented along with pointers to further materials and relevant information.

The paper reports also on the EKAW'14 tutorial on "Language Resources and Linked Data" that treated all the above concepts in a practical way. Pointers to the tutorial's instructional material and required software are also given, with the aim at helping developers and interested readers to acquire the basic mechanisms to contribute to the LLOD cloud with their own resources.

Acknowledgments. This work is supported by the FP7 European project LIDER (610782) and by the Spanish Ministry of Economy and Competitiveness (project TIN2013-46238-C4-2-R).

References

1. Archer, P., Goedertier, S., Loutas, N.: Study on persistent URIs. Technical report, December 2012
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal on Semantic Web and Information Systems (IJSWIS) **5**(3), 1–22 (2009)
3. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia - a crystallization point for the web of data. Web Semantics: Science, Services and Agents on the World Wide Web **7**(3), 154–165 (2009)
4. Chiarcos, C., Nordhoff, S., Hellmann, S. (eds.) Linked Data in Linguistics - Representing and Connecting Language Data and Language Metadata. Springer (2012)
5. Cimiano, P., Buitelaar, P., McCrae, J.P., Sintek, M.: LexInfo: A declarative model for the lexicon-ontology interface. Web Semantics: Science, Services and Agents on the World Wide Web **9**(1), 29–51 (2011)

[40] Compare to the 40 and 31 participants that followed the related tutorials at LREC and ISWC respectively. Notice, however, the smaller size of the hosting conference in the case of EKAW.

6. Eckle-Kohler, J., McCrae, J.P., Chiarcos, C.: LemonUby-A large, interlinked, syntactically-rich lexical resource for ontologies. Semantic Web Journal-Special issue on Multilingual Linked Open Data (2015)
7. Ehrmann, M., Cecconi, F., Vannella, D., McCrae, J.P., Cimiano, P., Navigli, R.: Representing Multilingual Data as Linked Data: the Case of BabelNet 2.0. In Proceedings of the 9th Language Resource and Evaluation Conference, pp. 401–408 (2014)
8. Gracia, J., Montiel-Ponsoda, E., Vila-Suero, D., Aguado-de Cea, G.: Enabling language resources to expose translations as linked data on the web. In Proc. of 9th Language Resources and Evaluation Conference (LREC 2014), Reykjavik (Iceland), pp. 409–413. European Language Resources Association (ELRA), May 2014
9. Hellmann, S., Lehmann, J., Auer, S., Brümmer, M.: Integrating NLP using linked data. In: Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J.X., Aroyo, L., Noy, N., Welty, C., Janowicz, K. (eds.) ISWC 2013, Part II. LNCS, vol. 8219, pp. 98–113. Springer, Heidelberg (2013)
10. Manola, F., Miller, E.: RDF primer. Technical report, W3C Recommendation (February 2004)
11. McCrae, J.P., Aguado-de Cea, G., Buitelaar, P., Cimiano, P., Declerck, T., Gómez-Pérez, A., Gracia, J., Hollink, L., Montiel-Ponsoda, E., Spohr, D., et al.: Interchanging lexical resources on the semantic web. Language Resources and Evaluation **46**(4), 701–719 (2012)
12. McCrae, J.P., Unger, C., Quattri, F., Cimiano, P.: Modelling the semantics of adjectives in the ontology-lexicon interface. In: Proceedings of 4th Workshop on Cognitive Aspects of the Lexicon (2014)
13. McCrae, J.P., Fellbaum, C., Cimiano, P.: Publishing and linking wordnet Dusing lemon and RDF. In: Proceedings of the 3rd Workshop on Linked Data in Linguistics (2014)
14. Miller, G.: WordNet: A Lexical Database for English. Communications of the ACM 38(11), November 1995
15. Montiel-Ponsoda, E., Aguado de Cea, G., Gómez-Pérez, A., Peters, W.: Modelling multilinguality in ontologies. In: Proceedings of the 22nd International Conference on Computational Linguistics, pp. 67–70 (2008)
16. Moro, A., Raganato, A., Navigli, R.: Entity Linking meets Word Sense Disambiguation: a Unified Approach. Transactions of the Association for Computational Linguistics (TACL) **2**, 231–244 (2014)
17. Navigli, R., Ponzetto, S.P.: BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence **193**, 217–250 (2012)
18. Unger, C., Bühmann, L., Lehmann, J., Ngonga Ngomo, A.-C., Gerber, D., Cimiano, P.: Template-based question answering over RDF data. In: Proceedings of the 21st International Conference on World Wide Web, pp. 639–648 (2012)
19. Vila-Suero, D., Gómez-Pérez, A., Montiel-Ponsoda, E., Gracia, J., Aguado-de Cea, G.: Publishing linked data: the multilingual dimension. In: Cimiano, P., Buitelaar, P. (eds.) Towards the Multilingual Semantic Web, pp. 101–118. Springer (2014)
20. Villazón-Terrazas, B., Vilches, L., Corcho, O., Gómez-Pérez, A.: Methodological guidelines for publishing government linked data. In: Wood, D. (ed.) Linking Government Data, ch. 2. Springer (2011)