

How to publish Linguistic Linked Data

Publishing data on the Web in a way that it can be easily consumed is a daunting task. The Web community has converged on the idea that Linked Data provides the best practices for publishing highly interoperable data. This card explains in detail what these best practices are.

Target audience: this card is aimed at data publishers who already have language resources published on the Web and want to go the next step and increase their interoperability.

Scope: this card provides clear steps one needs to follow in order to publish Linked Data on the Web.

1) RDF Conversion

Your data must be published using one of the many available RDF serializations. You can choose between **Turtle**, **JSON-LD**, **RDF/XML** and many others. All RDF formats are interoperable with one another, meaning one can automatically convert Turtle into JSON-LD. Here's an example on how to *triplify* language resources using Turtle:

Noun	Definition	Language
Riga	A port city on the Gulf of Riga...	English
Rīga	Rīga ir Latvijas galvaspilsēta un galvenais industriālais...	Latvian

Lexical entries for the city *Riga*, *Latvia* in two languages



```
bn:Riga_n_EN a lemon:LexicalEntry ;
  rdfs:label "Riga"@en ;
  lemon:language "en" .

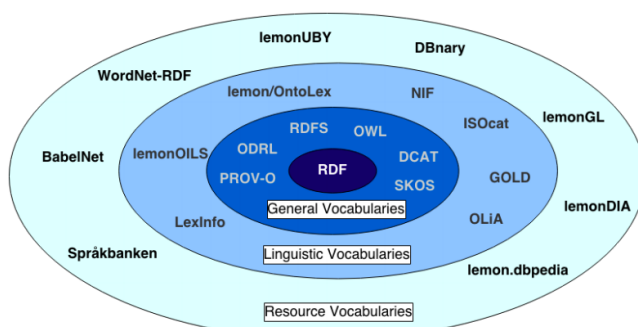
bn:Riga_n_LV a lemon:LexicalEntry ;
  rdfs:label "Rīga"@lv ;
  lemon:language "lv" .
```

RDF representation of the same entries using the RDF *Turtle* syntax

2) Data Modeling

Data should be modelled using well established ontologies and vocabularies. We identify three different types of vocabularies:

- **General vocabularies:** can be used not only for linguistic resources but also all kinds of resources. Ex: RDFS, OWL, SKOS, etc.
- **Linguistic vocabularies:** used to model language resources such as lexicons and corpora. Ex: *lemon*, OntoLex, NIF, etc.
- **Resource-specific vocabularies:** used for modeling information that is specific for a particular type of language resource. Ex: WordNet-RDF, BabelNet, DBnary, etc.



3) Linking Data

When there are similar resources in multiple languages, it should be possible to provide links between them to enable multilingual use. We consider the following options:

- **Interlingual identity links:** identity stating properties such as `owl:sameAs` or `skos:exactMatch` can be used to establish links between equivalent resources in different languages.
- **Interlingual soft links:** alternatively, weaker properties such as `rdfs:seeAlso` can be used, but, since these properties have no clear semantics, their value for applications is limited.
- **Linking to a common index:** finally, instead of establishing links directly, it may be preferable to establish indirect links by mapping to a common ontology such as BabelNet or DBpedia.

```
@prefix bn: <http://babelnet.org/rdf/> .

bn:Riga_n_EN a lemon:LexicalEntry ;
  rdfs:label "Riga"@en ;
  lemon:language "EN" ;
  lemon:sense <http://babelnet.org/rdf/Riga_EN/s00015564n> .

<http://babelnet.org/rdf/Riga_EN/s00015564n> a lemon:LexicalSense ;
  lemon:reference bn:s00015564n .

bn:s00015564n a skos:Concept ;
  skos:exactMatch lemon-Omega:OW_eng_Synset_39689, lemon-WordNet:wn30-
09013353-n, dbpedia:Riga .
```

Example of a Linked Data resource from BabelNet and how it is linked to three other resources, namely DBpedia, OmegaWiki and WordNet

4) Resolvable URIs

RDF requires URIs to identify things. The Linked Data principles go one step further and require the use of HTTP URIs (starting with `http://` or `https://`), so they can be **resolved** and return appropriate RDF information. Below we show an example of how a Linked Data URI is resolved and what data it returns.

```
$ curl -L -H "Accept:text/turtle" http://babelnet.org/rdf/Riga_n_EN
bn:Riga_n_EN a          lemon:LexicalEntry ;
  rdfs:label          "Riga"@en ;
  lemon:canonicalForm <http://babelnet.org/rdf/Riga_n_EN/canonicalForm> ;
  lemon:language      "en" ;
  lemon:sense         <http://babelnet.org/rdf/Riga_EN/s00015564n> ;
  lexinfo:partOfSpeech lexinfo:noun .
```

Resolving the `bn:Riga_n_EN` URI from BabelNet, returns information about it