
# D5.1 Report on Vocabularies for Interoperable Language Resources and Services

Author(s): Christian Chiarcos, Philipp Cimiano, Julia Bosque-Gil, Thierry Declerck, Christian Fäth, Jorge Gracia, Maxim Ionov, John McCrae, Elena Montiel-Ponsoda, Maria Pia di Buono, Roser Saurí, Fernando Bobillo, Mohammad Fazleh Elahi

Date: 2020-01-25

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825182. The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Union.

**H2020-ICT-29b**
**Grant Agreement No. 825182**
Prêt-à-LLOD - Ready-to-use Multilingual Linked Language Data for Knowledge Services across Sectors

# *D5.1 Report on Vocabularies for Interoperable Language Resources and Services*

Deliverable Number: D5.1
Dissemination Level: Internal (for preliminary version) / Public (for final version)
Delivery Date: 2019-11-15 (Preliminary version. Final version due 2019-12-31)

Version: 1.0
Author(s): Christian Chiarcos, Philipp Cimiano, Julia Bosque Gil, Thierry Declerck, Christian Fäth, Jorge Gracia, Maxim Ionov, John McCrae, Elena Montiel-Ponsoda, Maria Pia di Buono, Roser Saurí, Fernando Bobillo

**Document History**

| Version Date | Changes | Authors |
|---|---|---|
| **2019-10-22** | **Initial Document** | **Christian Chiarcos, Philipp Cimiano, Christian Fäth** |
| **Oct/Nov 2019** | **Partner contributions** | **all co-authors** |
| **2019-11-12** | **Consolidation** | **Christian Chiarcos** |
| **2019-11-15** | **Review (Draft 0.9)** | **Fernando Bobillo** |
| **Nov/Dec 2019** | **Web Service Metadata, Provenance** | **Mohammad Fazleh Elahi, Philipp Cimiano** |
| **2020-01-15** | **Consolidation** | **Christian Chiarcos** |

# Table of Contents

# D5.1 Report on Vocabularies for Interoperable Language Resources and Services

## 1. Overview

This document provides a survey over vocabularies for language resources and services and sketch necessary extensions and the expected contribution of the Prêt-à-LLOD project to their further development for phenomena currently not sufficiently covered. Future updates with respect to this will be documented within Task 5.4.

We focus on three main aspects of linguistically analyzed data

1. lexical-conceptual resources, i.e., repositories of terminology, lexical data, translation, and semantics,
2. linguistically annotated data, concerning linguistic analysis of textual or transcribed data, and
3. language resource terminology, i.e., linguistic data categories and metadata

For these areas, we describe representative vocabularies from the Linguistic Linked Open Data community (RDF-based vocabularies) as well as other approaches (e.g., ISO TC37 standards), we identify a number of gaps, and we describe ongoing efforts to address these gaps within the Prêt-à-LLOD project.

# 2. Terminology, Lexical Data, Translation, Semantics

In this section, we describe vocabularies for lexical-conceptual resources, i.e., repositories of terminology, lexical data, translation data and natural language semantics.

## 2.1 Existing Models and Standards

### 2.1.1 Terminology

#### The TermBase eXchange format (TBX)

TermBase eXchange (TBX) is an international standard (ISO 30042:2019)[1] for the representation of structured concept-oriented terminological data. Initially published by the Localization Industry Standards Association (LISA), it has been released under a Creative Commons license in 2011, when LISA ceased its operations.

The foundations for TBX have been established by three international standards: (i) TMF (ISO 16642:2003), which defines the structural metamodel for TBX and other TMLs (terminological markup languages); (ii) ISO 12620, which provides an inventory of data-categories for terminological data; (iii) MARTIF (ISO 12200:1999), which provides the basis for the core structure of TBX and the XML styles of its elements and attributes.

TBX provides an XML-based framework to manage terminology, knowledge and content, by means of several processes, such as analysis, descriptive representation, dissemination, and interchange (exchange).

The TBX framework is composed of two main modules: a core-structure module and an XCS (eXtensible Constraint Specification) module. The former includes high-level elements which are in correspondence with the TMF metamodel. The latter is based on a formalism for identifying a set of data-categories and their constraints. The core-structure module is defined in a DTD used together with an XCS file that applies additional data-category constraints.

Data-categories are the result of the specification of a given data field, e.g., part of speech, or grammatical number. In order to guarantee high interoperability, TBX provides a default set of data-categories that are commonly used in terminological databases. Data-categories can be implemented using either an attribute or the content of an element.

A data-category implemented using an attribute is a terminological data-category that is defined according to ISO 12620, such as */definition/*, and one that is specified as a value of the name attribute in the default XCS file.

[1] For this documentation we refer to the official documentation available at https://www.gala-global.org/sites/default/files/uploads/pdfs/tbx_oscar_0.pdf

A data-category implemented as the content of an element is a simple data-category, that is, one value of a closed set of values (picklist). These terminological data-categories are also documented according to ISO 12620.

The specification of the value of an attribute, the content of an element, or one or more structural levels, may be formalized through data-category constraints, which limit the application of a meta data-category, a core-structure module data-category that takes a type attribute and facilitates modularity. The default TBX data-categories and their constraints includes elements or attribute, implemented directly in the core-structure DTD, and specializations, e.g., concept relations, properties and description of terms, of the metadata-categories.

TBX is directly relevant to Prêt-à-LLOD, as many data sets to be transformed in Task 3.1 are encoded in this standard.

### Simple Knowledge Organization Scheme (SKOS) and SKOS-XL

The Simple Knowledge Organization System (SKOS) "is a common data model for sharing and linking knowledge organization systems via the Semantic Web"[2] and it "is an RDF vocabulary for describing the basic structure and content of concept schemes such as thesauri, classification schemes, subject heading lists, taxonomies, 'folksonomies', other types of controlled vocabulary, and also concept schemes embedded in glossaries and terminologies"[3]. Those two quotations are describing very well the scope and format of the SKOS W3C recommendation.

SKOS is based on the RDF vocabulary and it is also making use of RDF(S)[4]. RDF(S) introduced so-called "annotation properties", like *rdfs:label* or *rdfs:comment*. Those annotation properties have been introduced in the RDF(S) vocabulary in order to equip OWL[5] ontological elements, like classes, properties or instances, with additional metadata and also human readable descriptions of the modelled knowledge objects. SKOS introduces three additional annotation properties that can be considered as a specialisation of *rdfs:label* for addressing terminological purposes: *skos:prefLabel*, *skos:altLabel* and *skos:hiddenLabel*. The values of such annotation properties are literals, and have as such no ontological status and can thus not be designated by a URI and consequently can not be used as a subject or a predicate in RDF triples. SKOS alone would thus not allow us to formally state relations between the terms represented by the labels. Fortunately, the W3C community has proposed a remedy to this situation, and defined a corresponding recommendation called SKOS-XL.

SKOS-XL stands for "Simple Knowledge Organization System eXtension for Labels'", providing additional support for describing and linking label elements of knowledge systems[6] SKOS-XL is, in a sense, elevating the values of the *skos:prefLabel*, *skos:altLabel* and

[2] Quoted from https://www.w3.org/TR/skos-reference/skos-xl.html
[3] Quoted from https://www.w3.org/2009/08/skos-reference/skos.rdf
[4] RDF stands for "Resource Description Framework" and RDF(S) is adding a data model for the basic RDF vocabulary. See also https://www.w3.org/TR/rdf-schema
[5] OWL stands for "Web Ontology Language", a Semantic Web representation language for modelling knowledge. See also https://www.w3.org/OWL
[6] See also http://lov.okfn.org/dataset/lov/vocabs/skosxl

*skos:hiddenLabel* properties to the same level as concepts (or "objects") defined in the knowledge sources, supporting thus the cross-linking of labels or their linking to other formal objects. In SKOS-XL concepts and labels that describe them are the same type of object/entities to which a URI can be associated. Relations between SKOS-XL labels can thus be explicitly and formally defined. A *skos:Concept* can relate to a *skosxl:Label* object via a *skosxl:prefLabel*, a *skosxl:altLabel* or a *skosxl:hiddenLabel* property and users can define all types of relations between *skosxl:Label* objects. This way we can state explicit relations between labels (representing terms) within one classification scheme but also between two or more classification schemes. It is possible now to formally express within a classification system that a term is the translation or the abbreviation of another term. While the SKOS-XL vocabulary is representing an important improvement for establishing those types of relations between terms encoded in Semantic Web compliant data sets, it is still lacking the capability to effectively support the lexical description of such term. This was also one of the reasons why OntoLex-Lemon has been designed and developed in the context of a W3C Community Group. Declerck et al. (2018) describe the relation of SKOS to OntoLex-Lemon for a specific use case, but also investigates the use of SKOS-XL for a better integration of terminological and lexicographic data.

## 2.1.2 Lexical Data

### Lexical Markup Framework (LMF)

The Lexical Markup Framework (LMF) is an ISO standard (with number 24613:2008) that was designed and adopted in the context of the ISO TC37/SC4 committee on Language Resources. It is the result of a cooperation between the Machine-Readable Dictionary (MRD) and the Natural Language Processing (NLP) communities. LMF is a model for representing lexical resources, with a focus on lexical entries. The model itself was represented as a UML[7] diagram. The model is a very generic one, with few main elements (or "classes"), which are *Lexical Resource*, *Global Information*, *Lexicon*, *Lexical Entry*, *Lemma*, and *Word Form,* while specialized modules can be attached to (for example) the element *Lexical Entry*. This modular approach was responding to many needs in the MRD and NLP communities and the standard has been widely discussed and tested.

While the formative part of the standard is describing the model and represents it using the UML approach, the so-called informative part of the document is describing an XML serialization of the model. While the use of XML is not mandatory, it remained the usual serialization of the model, although the model has also been ported to RDF[8].

[7] UML stands for "Unified Modeling Language". See also https://en.wikipedia.org/wiki/Unified_Modeling_Language
[8] See http://www.cltl.nl/projects/previous-projects/cornetto-lmf-rdf

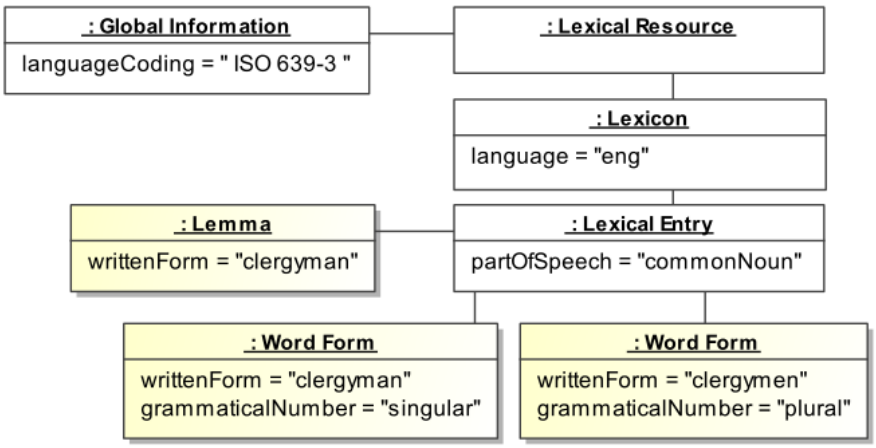



Fig. 1: UML diagram for an LMF entry
(taken from https://en.wikipedia.org/wiki/ Lexical_Markup_Framework)

LMF is in a way a starting point to the development of OntoLex-Lemon (see the section below), especially its modular approach was adopted by OntoLex-Lemon. The main differences between the two models are described in the specification document of OntoLex-Lemon, which we quote here (adding the sequence "[OntoLex-]" appropriately):

- "[OntoLex-]lemon defines the meaning of a term by reference to an ontology element defined by the OWL model.
- [OntoLex-]lemon provides a more compact description than LMF to describe the syntax-semantics interface.
- [OntoLex-]lemon relies on external category system and linguistic ontologies to describe linguistic properties of lexical entries instead of proposing an own category system.
- [OntoLex-]lemon does not include a module for describing inflectional morphology patterns (called intentional morphology in LMF). Further, it does not allow to define global constraints on the lexicon. This can be done using OWL axioms, but not in [OntoLex-]lemon itself." (https://www.w3.org/2016/05/ontolex/#lmf).

As presented in the next section, OntoLex-Lemon is representing its model using the RDF modelling language, allowing for a direct link to a standard serialization of RDF, be it Turtle, RDF/XML or the like. This represents a big advantage over the LMF modelling approach, which requires to map a UML diagram to an external serialization language. And last but not least, OntoLex-Lemon is perfectly suited for a direct use in the (Semantic) Web and for connecting lexical data to other knowledge data.

An interesting past initiative is the Cornetto project (Maks et al., 2013), in which Dutch lexical data was encoded using the XML serialization of LMF but the corresponding WordNet items were encoded using SKOS and RDF, while the OntoLex-Lemon model, described in the next section, can encode both types of data in the same format.

## OntoLex-Lemon

The lemon model, first proposed in (McCrae et al., 2012), has become the primary model for the representation of lexical data on the Semantic Web and has been further developed in the context of the W3C OntoLex Community Group[9]. After the lemon model was developed in the context of the Monnet project, it was decided that the further development of this model should take place within a forum as open as possible, which fortunately coincided with the creation of community groups for W3C. The community group structure provided mailing lists and wikis for discussion of the model and eventually led to the publishing of the model as a W3C Report (Cimiano et al., 2016) and as files in the W3C namespace.

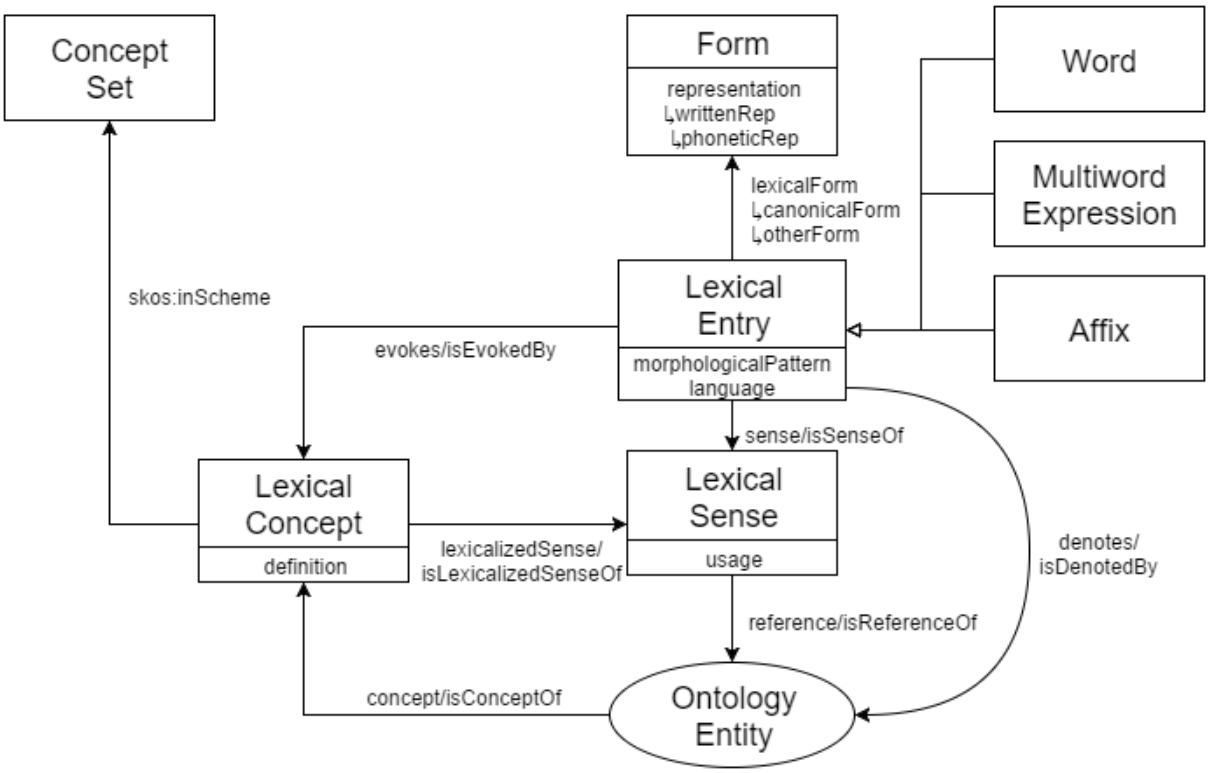



Fig. 2: OntoLex-Lemon core model

The model primary element is the lexical entry (see Figure 2 above), which represents a single word with a single part-of-speech and set of grammatical properties. This entry is composed of a number of forms and also composed of a number of sense which enumerate its meanings. The senses can be defined formally by reference to an ontology or informally by a lexical concept, which defines a concept in a non-linguistic and hence cross-lingual manner.

There has been much discussion of all aspects of the model, however the issue of semantics was of particular interest to the group and led to a major innovation in the introduction of a lexical concept, as a distinct element from the ontology reference. The formal distinction between these is that an ontology reference is an entity in an ontology, which the word denotes. As such, the (ontological) meaning of the question “When did Prince die?” could be understood with the ontology predicate deathDate as a reference, but the general concept of dying refers to an event rather than to a date. This also further

[9] See https://www.w3.org/community/ontolex/ for the W3C Community Group “Ontology Lexica”

extends the application domain of OntoLex-Lemon, from formal applications such as question answering and semantic parsing to the representation of general machine-readable dictionaries, including WordNet and digitized versions of existing dictionaries.

Thus, the OntoLex-Lemon model has continued to expand in its use cases and has been adopted in a variety of online dictionaries and this has provided a common interface to these dictionaries. Further, OntoLex-Lemon in the context of the WordNet Collaborative Interlingual Index (Bond et al., 2016), where the model is being used to provide a single interlingual identifier for every concept in every language.

Finally, the OntoLex-Lemon model is continuing to be developed and four areas have recently identified for extension: increasing support for use cases of the model in representing digitized dictionaries (lexicography module), the use of clear and defined data categories to improve interoperability (LexInfo revision), the development of a module for representing complex morphological patterns (morphology module), and extension to support the representation of frequency, attestation and corpus-based information within the lexicon. These developments should further increase the applicability and value of the model to more users.

### 2.1.3 Translation

#### Translation Memory eXchange (TMX) format

The Translation Memory eXchange format (TMX) is the de-facto standard in the field of automated translation and is widely used in computer aided translation (CAT) systems, as a means for exchanging translation memories, supporting thus the re-use of already translated language data in both human and machine translations. It is not a model (contrary for example to LMF and OntoLex-Lemon), but rather an XML specification. Language data contained in TMX datasets are also used as the basis for training statistical or neuronal machine translation systems, as this is for example for the eTranslation infrastructure of the European Commission[10]. And there are many aligned corpora available in TMX[11].

The XML specification includes a header and a body part. The header is foreseen for administrative information about authors or creation tools, date of the data, the administrative language of the data set and the source language, etc. The body part contains then the language segments in the source language and one or more target languages, as the example in Figure 3 shows:

[10] Some of the TMX datasets used for training the neuronal translation system of the EC are publicly available at the ELRC-SHARE portal (https://elrc-share.eu).
[11] So for example (subsets of) the Europarl corpus (http://opus.nlpl.eu/Europarl.php).

```
<tmx version="1.4">
  <header
    creationtool="XYZTool" creationtoolversion="1.01-023"
    datatype="PlainText" segtype="sentence"
    adminlang="en-us" srclang="en"
    o-tmf="ABCTransMem"/>
  <body>
    <tu>
      <tuv xml:lang="en">
        <seg>Hello world!</seg>
      </tuv>
      <tuv xml:lang="fr">
        <seg>Bonjour tout le monde!</seg>
      </tuv>
    </tu>
  </body>
</tmx>
```

Fig. 3: Simple example for a TMX document, with one entry containing an English source and a French target segment (taken from https://en.wikipedia.org/wiki/Translation_Memory_eXchange)

TMX datasets are very relevant to the pilots of Prêt-à-LLOD, as one can focus on aligned multilingual data in a specific domain. But in order to publish efficiently such data in the Linked Data cloud, the TMX encoding needs to be transformed into a Linked Data compliant format. Additionally, TMX datasets do not include lexical or syntactic information, so that the text segments should be processed by an NLP pipeline in order to get the needed lexical markup (for example using OntoLex-Lemon, also considering its VarTrans[12] module for formally marking the translation relations between the words) and syntactic annotation (using for example NIF), in order to be made operable in the pilots of Prêt-à-LLOD.

### OntoLex-VarTrans

OntoLex-VarTrans is one of the modules of the Ontolex-Lemon model specifically intended to account for lexico-semantic relations between entries, senses, or mental concepts. VarTrans stands for variation and translation, which means that the types of lexico-semantic relations currently considered by the module are relations between terms in the same language (term variants) or between terms in different languages (i.e., translations).

In the VarTrans module, lexical relations represent the relation between two lexical entries whose surface forms are related grammatically, stylistically, or by linguistic economy. These are the types of relations between adjectives and adverbs (quick vs. quickly), abbreviation or acronym relations, or relations between terms following a different morphosyntactic pattern (agroindustry vs. agricultural industry).

Sense relations, on the contrary, represent the relation between two lexical senses or lexical concepts. Typical examples of sense relations that can be represented here are hyperonymy and hyponymy relations, synonymy, or antonymy relations. However, and depending on the type of ontology being lexicalised, when dealing with specialised language we may need to

[12] See next section and https://www.w3.org/2016/05/ontolex/#variation-translation-vartrans for more details.

specify the synonymy relation into more specific types of relations between terms, what we refer to as terminological relations or term variants. Such variants vary along dimensions that are pragmatically caused, such as the geographical context (zip code vs. postal code), chronological variants (tuberculosis vs. phthisis), formal and informal situations (cancer vs. neoplasm), or dimensional variants (genetic engineering vs. genetic manipulation).

When dealing with the relations between lexical entries or lexical senses across languages, we refer to them as translations in the broader sense of the word, i.e., from designations in different languages that denote the same ontology concept, to designations that "pragmatically work" as equivalents or counterparts, but denote different ontology entities.

As in the case of term variants, should we want to specify the type of translation relation, we could also do so by means of the category property, although the Ontolex-Lemon model does not further specify the types of translation relations. For this purpose, however, we can refer to an external categorization proposed by the main contributors to the VarTrans module (http://linguistic.linkeddata.es/def/translation-content/index.html).

### 2.1.4 Relational Semantics (Frame Semantics)

#### Classical digital resources for frame semantics

Frame semantics has been an area of intense research since Fillmore's seminal "Case for Case" article (Fillmore, 1968), and numerous digital resources for frame semantics have subsequently emerged, most notably FrameNet (Baker et al., 1998) and PropBank (Kingsbury and Palmer, 2003) (other specifications do exist, but FrameNet and PropBank are more representative in that their specifications have been applied to several other languages beyond English): FrameNet is an inventory of frames, i.e., predicates, their roles and potential fillers, and constraints for those, coupled with lexicalization preferences and subsequently augmented with annotations in actual text. PropBank is an annotation effort that develops a frame inventory as a means to annotate textual data. Both differ in philosophy and granularity, but are nevertheless closely interrelated and complementary resources. Unfortunately, their respective data models and formats are quite different, so that harmonization between both resources could only be implemented by untyped hyperlinks (the Unified Verb Index, http://verbs.colorado.edu/verb-index/index.php, Palmer, 2009). This mapping is informative, but incomplete and not machine-readable, as it is implemented on the level of human-readable visualizations (websites) rather than machine-readable web resources.

More recent efforts to integrate both resources with each other and related resources (VerbNet, NomBank, BabelNet, etc.) have thus been developed on the basis of Linked Data principles and technology. At the same time, we are faced with a multitude of proposals for vocabularies for this purpose, so that the desideratum is less to develop novel or more adequate vocabulary, but rather to harmonize or synthesize existing proposals.

### PreMon ontology

The PREdicate Model for ONtologies (PreMon, Rospocher et al., 2019)[13] is an ontology that extends the lemon model to provide for the representation of predicate models and their mappings. PreMon supports the representation of predicate models such as PropBank, NomBank, VerbNet and FrameNet. PreMon provides an OWL ontology for modelling semantic classes (i.e., verb classes, rolesets, or frames) with their roles, mappings across different predicate models and to ontological resources, and annotations, based on OntoLex-Lemon. For this, the model extends lemon by introducing classes pmo:SemanticClass and pmo:SemanticRole. pmo:SemanticClass homogeneously represents the semantic classes from the various predicate models. Mappings are explicitly represented as individuals of class pmo:Mapping, and can be seen as sets of (or n-ary relations between) either (i) pmo:Conceptualizations, (ii) pmo:SemanticClasses, and (iii) pmo:SemanticRoles, with role mappings anchored to conceptualization or class mappings via property pmo:semRoleMapping. Structurally, a pmo:Conceptualization can be seen as the reification of the ontolex:evokes relation between ontolex:LexicalEntry and ontolex:LexicalConcept. Semantically, it can be seen as a very specific intensional concept (among many, in case of polysemy) evoked by a single ontolex:LexicalEntry, which can be generalized to a ontolex:LexicalConcept when multiple entries are considered but with a possible loss of information that prevents precise alignments to be represented. Besides the core PreMon vocabulary[14], there are extensions to represent predicate models in FrameNet, Propbank and VerbNet.

### Framester by STLab

Framester (Gangemi et al., 2016) is a linked data resource that acts as a hub between FrameNet, WordNet, VerbNet, BabelNet, DBpedia, Yago, DOLCE-Zero, as well as other resources. Framester is not only a strongly connected knowledge graph, but also applies a rigorous formal treatment for Fillmore's frame semantics, enabling full-fledged OWL querying and reasoning on a large frame-based knowledge graph.

Following frame semantics, which is a development of case grammar and relates linguistic semantics to encyclopaedic knowledge, Framester describes the frame evoked by a single word. The underlying idea is allowing to formalize the semantic frame of encyclopaedic meaning, evoked or activated by a word and related to the specific concept which the word refers to. Words are not only the expression of individual concepts, but also the description of a certain perspective in which the frame is viewed.

Framester core maps WordNet, BabelNet, VerbNet and FrameNet expanding them to other linguistic resources transitivetely. It features a subsumption hierarchy of semantic roles, namely frame elements and generic roles on top of frame-specific roles.

The core schema for Framester can be found at: https://w3id.org/framester/schema/. Framester has been released in version 3.0[15]. Framester can be queried via a SPARQL[16] endpoint and also features an Word-Frame Disambiguation API[17].

[13] http://premon.fbk.eu/
[14] http://premon.fbk.eu/ontology/core
[15] https://github.com/framester/Framester

### Rich Event Ontology (REO)

The Rich Event Ontology (Brown et al., 2017) provides an independent conceptual backbone to unify existing semantic role labeling (SRL) schemas and augment them with event-to-event causal and temporal relations. By unifying the FrameNet, VerbNet, Automatic Content Extraction, and Rich Entities, Relations and Events resources, the ontology serves as a shared hub for the disparate annotation schemas and therefore enables the combination of SRL training data into a larger, more diverse corpus. By adding temporal and causal relational information not found in any of the independent resources, the ontology facilitates reasoning on and across documents, revealing relationships between events that come together in temporal and causal chains to build more complex scenarios.

Intended as a resource for a wide range of tasks, the Rich Event Ontology (REO) has been designed to encompass both meta-level concepts in its upper level and many general domains in its mid level. REO has been implemented in OWL, which allows for easy extension with more detailed, domain-specific ontologies. The main reference ontology now encompasses 161 classes and 553 axioms. Including the lexical resource ontologies and the linking models (described in detail in sections 2.5 and 2.6) in these counts brings the totals to 3,065 classes and 60,531 axioms, as well as 16,005 individuals representing the vocabulary (unique lemmas) of event denotations.

## 2.2 New or Emerging Models and Standards

### 2.2.1 Current Shortcomings and Desiderata

We have shown representative vocabularies for lexical-conceptual resources that have been established in the community, and that provide the basis for more recent RDF vocabularies that adequately represent lexical-conceptual data in a LLOD-compliant way. Although these RDF vocabularies are in general as adequate for modelling lexical data as their predecessors that used other formalisms, we can observe the following gaps:

- Improvements in lexicographic vocabulary in OntoLex-Lemon

  The early development of OntoLex-lemon has been driven by technical applications, mostly. Lexical knowledge, however, resides with the lexicographer, not with the engineer, so it is necessary to acknowledge their needs and habits, and in particular, their terminology, in order to facilitate the production of ready-to-use multilingual lexical data in a LLOD-compliant way.

  Lexicographic resources (e.g. dictionaries, lexica, terminologies) provide lexical data following different formats, annotation schemes, organization and criteria. However, a full conversion of the data given in the dictionary record to RDF is not a trivial task in itself. While the OntoLex-Lemon core and the various modules provide elements to

[16] https://w3id.org/framester/sparql
[17] https://w3id.org/framester

account for lexical description in a great extent, this description does not always fully reflect the information provided in a lexicographic resource: there is not always a 1:1 match between a dictionary element and an OntoLex element (e.g. a dictionary entry and an ontolex:LexicalEntry), and structural decisions grounded on morphological or semantic relations (e.g. groupings of forms, nesting of senses, etc.) are not representable with the OntoLex core on its own either. In addition, some annotations and data commonly offered in dictionary entries (e.g. regarding the morphosyntactic features of an entry when used in a specific sense) or usage examples, were not addressed by the OntoLex core or the various modules either. However, with OntoLex being descriptive and not prescriptive, an extension to better capture the dictionary representation in accordance with its original conception was called for. In turn, this extension could serve as a bridge between other formats used to encode lexicographic data and OntoLex-Lemon.

- Limited coverage of morphology in OntoLex-Lemon

  As electronic dictionaries are increasingly being used for natural language processing applications, morphological data needs to be provided in a way that can be readily processed by machines. However, there is a considerable amount of morphological information not straightforwardly representable as LLOD within the current model: distinctions between derivational and inflectional morphology, allomorphy, suppletion, simulfixes and transfixes, and information on morphological patterns. In addition, the description of morphological information in most traditional dictionaries is limited to the list of the word forms that allow users to identify the morphological pattern to which the entry adheres, and hence generate the paradigm by themselves. Following this, word-forms that can be formed regularly are not listed. Moreover, the description of these “reduced” inflection lists is often minimal on the assumption of users being familiar with the lexicographic tradition of the object language. Models compliant with OntoLex-Lemon that account for this information and that will allow for the future the automatic derivation of forms on the basis of the represented morphological patterns are lacking.

- Support for cross-resource linking between lexical data and corpus-based information

  Traditional data formats took a focus on a specific class of resources, and provided a solution for these, but only for these. In the context of Linked Data, data is no longer seen in isolation, but its potential for integration and synergies is increasingly emphasized. While RDF technology allows to provide flexible, and semantically typed links between *any* kind of resources, the usability and reusability of these links require novel, widely-used vocabulary elements. Here, we focus on links between lexical-conceptual resources (as formalized by OntoLex-lemon) and corpus information (cf. Sect. 3), i.e., frequency information, attestations and other data derived from corpora. Prêt-à-LLOD initiated and contributes to the development of such vocabulary as a module of OntoLex-lemon, and thus, backed by a significant user community, the current users of OntoLex-lemon.

- Standard vocabulary for relational / frame semantics

  Above, we discussed several vocabularies for representing relational / frame semantics in RDF and/or OWL. The desideratum here is not so much the development of a novel LLOD vocabulary for the purpose but a selection (or harmonization) process among the existing specifications. Within Prêt-à-LLOD, such a selection or harmonization process may be initiated. As a preliminary finding we observe that most RDF vocabularies for semantic frames agree that semantic frame( instance)s are defined as *ontolex:Concept*, not as *ontolex:Sense* nor as an external ontology element. This observation provides us with a convenient technological bridge between Ontolex-lemon and various vocabularies for frame semantics.

- Conventions for representing the uncertainty of semantic relations

  In linguistics, lexical semantic relations are commonly understood as relations holding between lexical elements (or lexemes) based on what they mean. Well-studied ones are synonymy, hyponymy and its opposite hypernymy, or meronymy. Lexical semantic relations can also be considered to apply cross-lingually, as is the case with translations, and they can acquire a broader meaning, including relations between the same lexical item as encoded in different resources. However, up to date there is not a mechanism to capture an uncertainty degree in lexical semantic relations. For example, we might be interested in translations that are imprecise or partially true, e.g., a Spanish "siesta" is slightly different than a "nap", so the translation holds to some degree of truth. Similarly, the definitions for an equivalent sense of a lexeme in two different dictionaries may map only to a certain extent, for example due to differences in editorial criteria on how to split word meaning into senses. In addition, there could be a term in a source language which can be translated perfectly as another term in a target language, but we are not sure if the translation is correct, i.e., if it is the right one. For example, the Spanish term "primo" has two senses and can be translated into English either as "prime" or as "cousin". This could be the case if we use an automatic software (e.g., Google Translate) to compute the translation of a term. In such cases, we might want to associate a confidence degree to the translation. A comparable situation could hold with a system to automatically link definitions for the same words in two different dictionaries.

### 2.2.2 Lexicog Module

The *lexicog* model[18] is the module for Lexicography of OntoLex-Lemon. As Figure 4 shows, it revolves around two basic layers, the structural and the lexical layer, with the lexical being represented mainly with OntoLex elements. The notion behind this separation is grounded on the assumption that the same lexical elements can be described differently in different sources, and what the lexicographic resource presents is a description of these lexical elements, arranged in a specific manner, i.e., a "view" on the lexicon.

[18] Available at https://www.w3.org/ns/lemon/lexicog. Module Specification published at https://www.w3.org/2019/09/lexicog/

The core elements of *lexicog* are lexicog:LexicographicResource, lexicog:Entry, and lexicog:LexicographicComponent, along with the properties lexicog:entry and lexicog:describes. The class lexicog:LexicographicResource is intended to represent the dictionary (prior to the conversion to RDF), that is, the collection of dictionary entries originally provided in the resource, which stems from the headword selection process. These are represented with lexicog:Entry and grouped in the dictionary through lexicog:entry. As there is not always a 1:1 match between a dictionary entry and an ontolex:Entry (for instance, a single dictionary entry may describe a lexeme that takes different parts of speech and thus corresponds to more than one ontolex:LexicalEntry), this difference also extends to the whole dictionary, and distinguishes a lime:Lexicon from a lexicog:LexicographicResource.

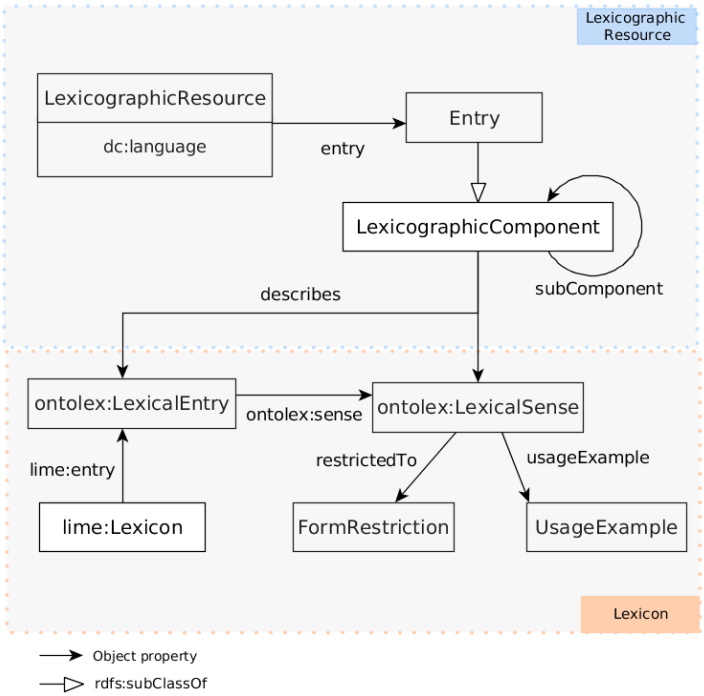



Fig. 4: The OntoLex lexicography module (*lexicog*). The upper part accounts for structures common in lexicographic resources, and the bottom part describes the lexicon, i.e., the elements describing the lexical information.

The class lexicog:LexicographicComponent serves a structural function and its instantiations act as “containers” of OntoLex elements that can be arranged, grouped or ordered in the same fashion as in the original resource. The property lexicog:describes links them to the actual lexical content captured with such OntoLex elements. The sub-sense hierarchy can be reflected with the use of lexicog:subComponent, while the ordering of senses is encoded through container membership properties (rdf:_1, rdf:_2, etc.). Lastly, some classes and properties of the module address the lexical layer and aim to represent examples of use of a lexical sense (lexicog:UsageExample) as well as morphosyntactic features of an entry when attested in one of its specific senses (lexicog:FormRestricion).

## 2.2.3 Ontolex Module for Morphology

There is substantial heterogeneity across dictionaries in the amount and type of morphological data provided in the dictionary entry (differences based on lexicographic tradition and approaches to lexicography (cf. Alsina and DeCesaris,1998; Swanepoel, 2015), which in turn leads to a heterogeneous landscape when it comes to analysing the morphological description provided in lexicographic resources.

The Morphology module of OntoLex-Lemon, which is currently under development[19], aims to account for a wide range of morphological information found in these dictionaries, e.g.: distinctions between derivational and inflectional morphology, allomorphy, suppletion, simulfixes and transfixes, and information on morphological patterns. The scope of the module is divided into two main parts: (i) morphological derivation which occurs on the lexical entry level and (ii) inflection, decomposition on the form level. Additionally, the module provides a mechanism for word-form generation using representations of paradigms and morphophonological transformations. This allows to provide morphological data that is available in dictionaries as morphological rules instead of lists of inflected forms.

Figure 5 shows the current state of the module. The core element for derivation is either a ontolex:LexicalEntry or its subclass ontolex:Affix (for cases where an element cannot be an independent entry). For combining derivational parts, OntoLex-decomp module is reused. Each part is represented by an instance of decomp:Component which corresponds to an instance of morph:Morph via a decomp:correspondsTo property. The links between a lexical entry and lexical entries derived from it can be established in two ways: either directly with a property morph:derivationalRel or with an instance of the class morph:DerivationalRelation which connects to two lexical entries with vartrans:source and vartrans:target properties.

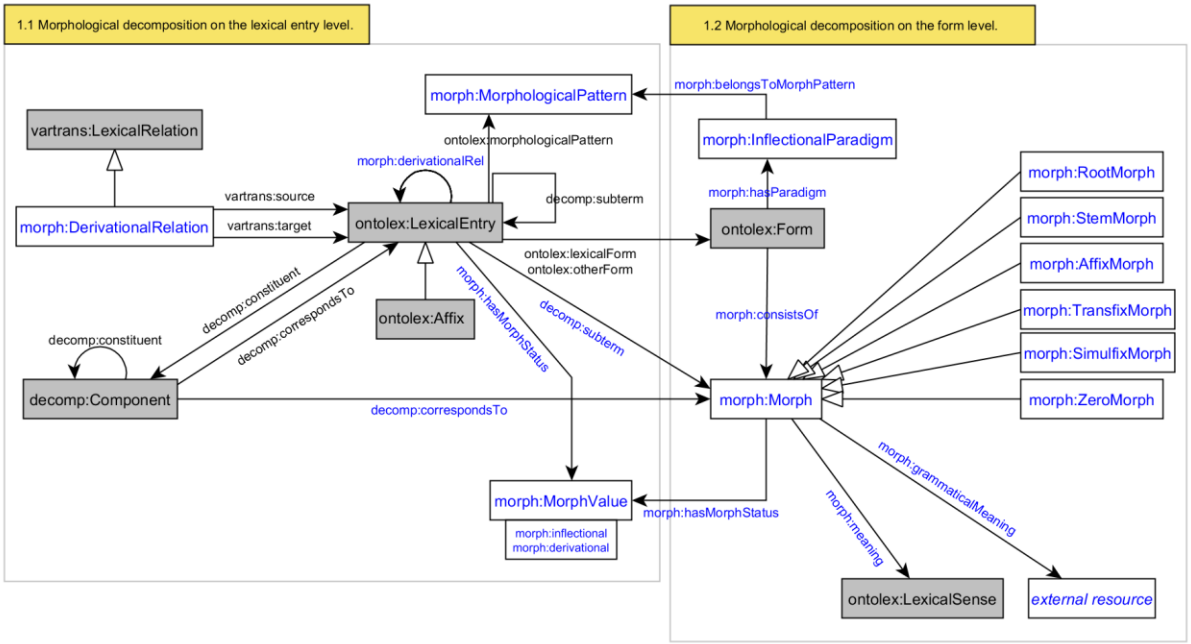



Fig. 5: The OntoLex morphology module. The left part accounts for representing derivation processes, the right part is used for representing inflection.

[19] https://www.w3.org/community/ontolex/wiki/Morphology

The former is a very generic statement but one that is often found in lexical or dictionary data. The latter explicitly interlinks the source lexical entry and the target lexical entry for which a unique derivational relation holds. For modelling inflection, an ontolex:Form class is instantiated connected to all the affixes (instances of morph:Morph) with the morph:consistsOf property.

### 2.2.4 OntoLex Module for Frequency, Attestations and Corpus Information (OntoLex-Frac)

OntoLex-Frac is the OntoLex module for frequency, attestation and corpus information currently being developed by Prêt-à-LLOD contributors together with the OntoLex community. Its development is motivated by requirements of computational lexicography, digital philology, and language technology.

The module is targeted at complementing dictionaries and other linguistic resources containing lexicographic data with a vocabulary to express (1) corpus-derived statistics (frequency and co-occurrence information, collocations), (2) pointers from lexical resources to corpora and other collections of text (attestations), (3) the annotation of corpora and other language resources with lexical information (lemmatization against a dictionary), and (4) distributional semantics (collocation vectors, word embeddings, sense embeddings, concept embeddings).

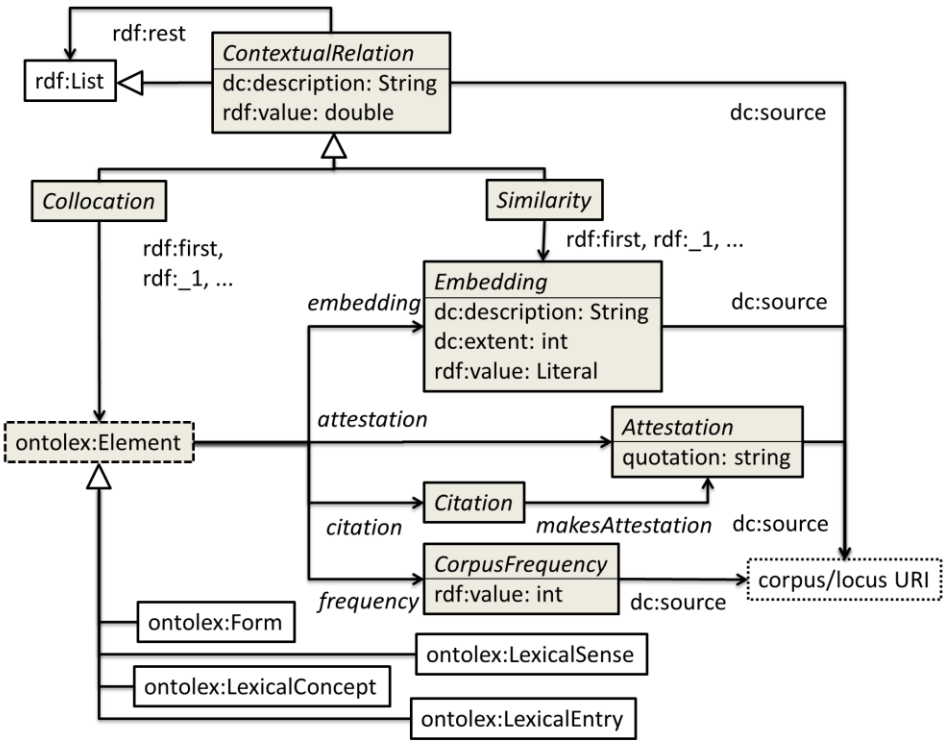



Fig. 6: The OntoLex Frac module. Draft November 2019, from https://acoli-repo.github.io/ontolex-frac

Frac development formally started with presentations by Prêt-à-LLOD contributors at the OntoLex Face-to-Face meetings in Leiden, Oct 2018 and Leipzig, May 2019, and is since conducted by means of bi-weekly calls and meetings since then. Figure 6 shows the current state of the module. The treatment of attestation information is to be considered to be stable, other parts of the module are still under development. Given the current progress rate, we expect to produce a consolidated vocabulary at the end of the Prêt-à-LLOD project.

### 2.2.5 Addressing Semantic Gaps in Relational Semantics

Most existing semantic representation models address lexical semantic aspects, which capture the underlying predicate-argument structure, without providing elements from logical semantics, which can be described as truth-conditional semantics and model-theoretic semantics. The emerging need is formalizing propositions, as idealised sentence suitable for logical manipulation, so that the meaning of the various parts of the propositions are given by a group of interpretation functions which license important inferences.

The main goal for emerging models should be providing a description for combining lexical and logical aspects in order to integrate typing predicates into the existing models and to model ambiguous predicates. In fact, as described by (Berant et al., 2011), different type signatures of the same predicate have different meanings, but given a type signature a predicate is unambiguous, and may reflect a distinction in the semantics that is not always obvious in the syntax. The representation of arguments to induce n-ary relations should allow to create a separate predicate for each pair of arguments of a word, furthering generalizations and supporting formal semantics for logical operators within linguistic theories.

A preliminary outcome of this discussion is a tentative recommendation for one particular candidate vocabulary introduced above. This discussion will be continued in exchange with the communities involved. For the moment, we express a preference for the PreMon vocabulary, as its development seems to be well-coordinated with the development of OntoLex-Lemon.

### 2.2.6 Modelling for Fuzzy Sense Relations

The aim of the extension of lemon for fuzzy sense relations is to allow to assign an uncertainty degree to lexical semantic relations. We propose the use of different formalisms to deal with different types of degrees of uncertainty (Lukasiewicz et al., 2008).

- On the one hand, fuzzy logic can manage statements with a degree of truth associated, expressing the extent to which the event described by such statement holds in the world. In the specific case of lexical semantic relations, this approach allows us to model the degree of semantic overlap between two terms in different languages, or between two senses in separate dictionaries. Here, a fuzzy degree 1 indicates full overlap, a degree 0 means no overlap at all, and an intermediate value

means that there is a partial overlap, which can happen because of a different sense granularity (where one definition has a wider denotation than the other one) or different sense boundaries (where both definitions share some part of their denotation but neither of them fully encloses the other one).

- On the other hand, possibilistic or probabilistic logics can manage confidence degrees. That is, the degree of knowledge about the certainty of the event. Now, there are several worlds, but we are not sure which is the right one.

To model uncertainty in Lemon, we start by defining a property semanticRelationDegree. Its domain is the class SenseRelation and its range are the decimal numbers in [0,1]. We propose to build a hierarchy of subproperties of semanticDegree to support different uncertainty types, as shown in Figure 7. Finally, we propose to extend the syntax of Lemon so that we can attach to a lexical relationship between senses a numerical degree in [0,1] via a subproperty of semanticDegree. For instance, we can add a fuzzy degree to the translation involving two entities example:siesta and example:nap as follows:

```
example:siesta a ontolex:LexicalEntry ;
               ontolex:sense example:siesta_sense .
example:siesta_sense ontolex:reference <http://dbpedia.org/ontology/Nap> .
example:nap a ontolex:LexicalEntry ;
           ontolex:sense example:nap_sense .
example:nap_sense ontolex:reference <http://es.dbpedia.org/resource/Siesta> .
example:trans a vartrans:Translation ;
      vartrans:fuzzyDegree 0.95 ;
      vartrans:source example:siesta_sense ;
      vartrans:target example:nap .
```

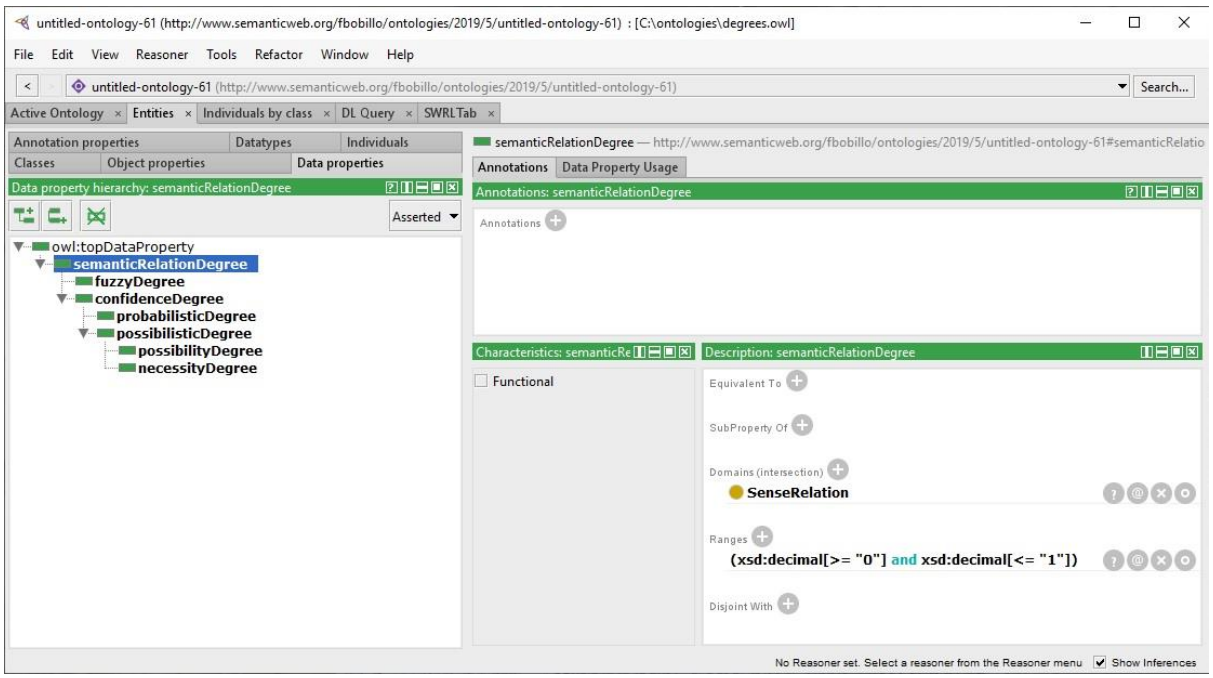



Fig. 7: hierarchy of subproperties to represent uncertainty degrees

The default value of the semantic degrees is 1, making our extension backwards compatible. Therefore, if the value is 1, there is not need to represent it explicitly.

A common problem when managing uncertainty is how to obtain the concrete values of the degrees. A first option is to ask a human expert, or a group of them, to assign the values. In our particular case, the proportion of lexical semantic relations with a confidence degree seems to be very small, so this could be a feasible solution. Another option is to use some (semi)automatic machine learning procedure to obtain the degrees from examples.

# 3. Linguistic Annotation

By linguistic annotation, we mean the annotation of textual or transcribed data with linguistic features. This does overlap with lexical data (e.g., corpus-derived information in lexical resources, Sect. 2.2.4 above; conjoint development of frame inventories and frame annotations in Sections 2.1.4 and 2.2.5 above), but in Sect. 2, this is described from the perspective of lexical resources, here from the perspective of (annotating) primary data with linguistic (e.g., lexical) information.

Existing formats are illustrated using the following example, slightly simplified from a clause from the OntoNotes corpus https://catalog.ldc.upenn.edu/ldc2013t19, file wsj-0655:

*James Baker ... told reporters Friday: "I have no reason to deny reports that some Contras ambushed some Sandinista soldiers."*

We focus on two, partially overlapping, aspects of linguistic annotation:

- How to anchor annotations in textual (or other) data
- How to model the structure of linguistic annotations

A third aspect of interoperable linguistic annotation, the formalization of linguistic data categories (that can be used for annotation) is discussed in Sect. 4, as it is not specific to linguistic annotation but can also be used for formalizing, e.g., linguistic features of lexical-conceptual resources.

## 3.1 Existing Models and Standards

### 3.1.1 Annotating Textual Data

#### Text Encoding Initiative, Proposal 5 (TEI P5)

The Text Encoding Initiative (TEI, http://tei-c.org) is the authoritative body that develops and maintains an XML-based interchange format for textual data, in particular for the electronic edition of printed (or printable) publications. Beyond its historical focus on literary science and linguistics, the current edition of the TEI guidelines, P5 (proposal 5), represents a de facto standard for electronic editions and the philologies. The TEI vocabulary provides hundreds of elements and attributes for the semantic and structural markup of electronic text.

The TEI aims to provide a compromise between a formal description of layout elements (e.g., italics) and their abstract function (e.g., emphasis). Its markup elements are given interpretable names, but the provided definitions are informative only, not normative, as the TEI standardizes only their form and structure. Accordingly, the TEI guidelines are traditionally implemented as ODD ("one document does it all") projects and validated by a

set of modular DTDs, resp. RelaxNG schemas derived from the ODD source. For practical applications, the TEI takes a text-driven approach: the form, content and structure of the underlying text is preserved, and are enriched by markup elements. A TEI document will thus thus always include the original document (text and/or layout), as in the example of syntactic annotations below.

```
<s type="sentence"> 2 <cl ana="#S">
<phr ana="#NP-SBJ">
<w ana="#NNP">James </w>
<w ana="#NNP">Baker </w>
    </phr>
    <phr ana="#VP">
    <w ana="#VBD">told </w>
<phr ana="#NP">
    <w ana="#NNS">reporters </w>
    </phr>
    <phr ana="#NP-TMP">
    <w ana="#NNP">Friday </w>
    <w ana="#colon">: </w>
    ...
    </phr> </phr>
    </cl>
    </s>
```

The TEI P5 guidelines provide generic data types for many forms of linguistic annotation, including elements for orthographic sentences (<s>), grammatical words (<w>) and grammatical phrases (<phr>), as well as attributes for their respective type (@type), interpretation (@ana) and identification (@xml:id).

### Standoff annotation and RCF 5147

Documents in the web come in various forms, and often, it is not possible to embed metadata and annotations directly into them, e.g., because the annotator is not the owner of the document and distributing a local copy may be restricted. Standoff formalisms support the physical separation of annotated material and annotations. Various standoff formalisms have been developed (e.g., also in the TEI), but here, we focus on more recent developments that implement standoff annotation by means of web standards. The technological basis for these is to provide URIs to sections or strings in a document. For plain text documents, this is provided by RFC 5147.

RFC 5147 (https://tools.ietf.org/html/rfc5147) defines an extension of earlier specifications for the text/plain MIME type. In general, URI fragment identifiers extend document URIs with a local name separated from the document URI using a hash sign (#). RFC 5147 provides a simple offset mechanism to address strings, i.e., sequences of characters, in a web document as follows:

- Character Position: A character offset starting from the beginning of the document, defining an empty string at a particular position in the document. For James Baker from the example above, we arrive at the following position URI:

```
https://catalog.ldc.upenn.edu/docs/LDC95T7/
raw/06/wsj_ 0655.txt#char=19
```

- Character Range: A consecutive sequence of characters with a particular start position and a particular end position, both defined as character offsets:

```
https://catalog.ldc.upenn.edu/docs/LDC95T7/
raw/06/wsj_ 0655.txt#char=19,30
```

- Line Offsets: Analogously to character offsets, a line offset refers to the number of lines (resp., line separators) before the designated position. The following example refers to the first line in the document:

```
https://catalog.ldc.upenn.edu/docs/LDC95T7/
raw/06/wsj_ 0655.txt#line=0
```

The text scheme is optionally followed by an integrity check, i.e., a length specification or an MD5 value:

```
...#char=19,30;length=12
...#char=19,30;md5=67f60186fe687bb898ab7faed17dd96a
```

Furthermore, a character encoding can be defined:

```
...#char=19,30;length=12,UTF-8
...#char=19,30;,UTF-8
```

Originally, RFC 5147 has been developed for highlighting strings in web documents. Aside from this application, RFC 5147 had a considerable impact on the language resource community, where its specifications have been extended for other MIME types and represent the basis for all URI schemes for strings, including NIF (see below), NAF (http://wordpress.let.vupr.nl/naf/), and LIF (https://wiki.lappsgrid.org/interchange/) – which are designed to address strings in character streams regardless of MIME type declarations.

### NLP Interchange Format (NIF 2.0)

The NLP Interchange Format (NIF, https://persistence.uni-leipzig.org/nlp2rdf/) is an RDF/OWL-based format designed to combine NLP tools in a flexible, light-weight fashion. NIF provides a way to map the annotations of two or more NLP pipelines into a common representation and to integrate them seamlessly on the basis of RDF technology, and in particular, reference to the same string URIs.

NIF includes the following core components:

- URI schemes to refer to strings in documents and to add annotations to such URIs. This includes RCF 5147 URIs and string URIs for other MIME types,
- OWL-based vocabulary to express relations between string URIs, and
- vocabulary extensions to represent frequent types of annotations in common NLP pipelines.

The core of NIF consists of a vocabulary for addressing arbitrary character sequences by RDF URIs to which linguistic annotations can be attached in a flexible fashion. By reference to a common pool of URIs, resp., by means of a mapping of annotated text data to a NIF representation, annotations from different NLP tools can be aggregated easily.

The NIF 2.0 Core ontology is shown in Figure 8.

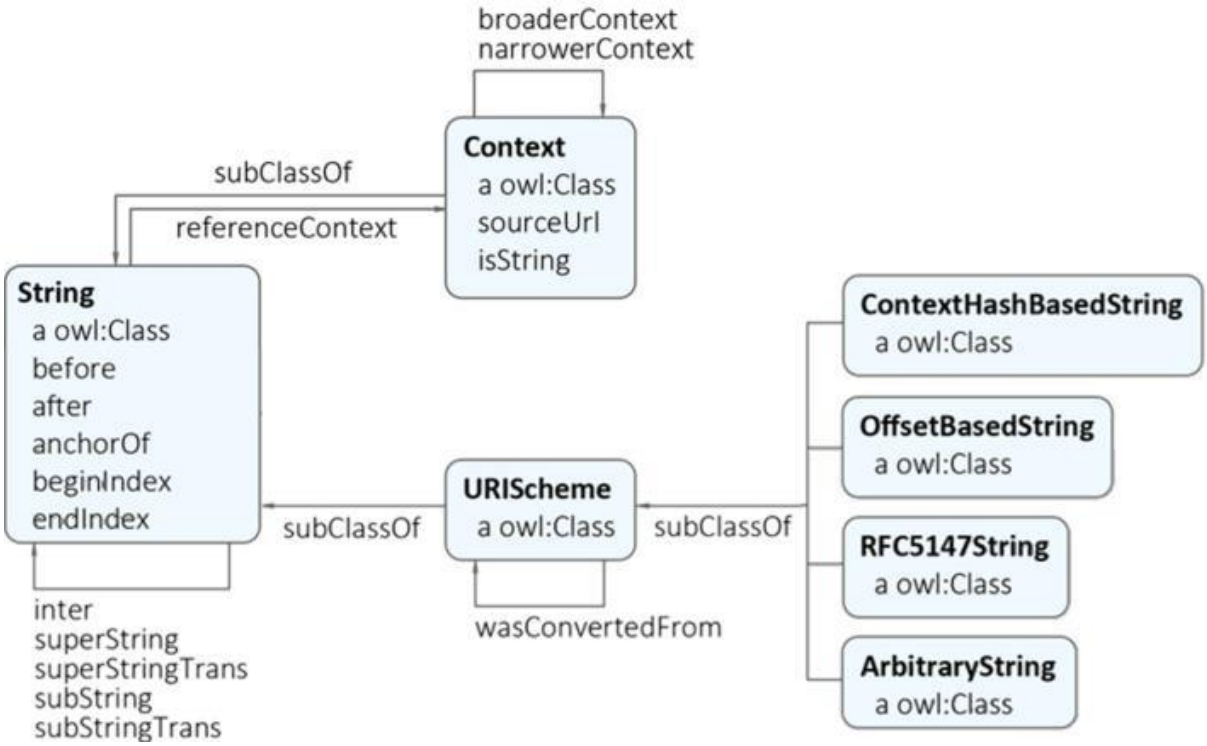



Fig. 8: The NIF 2.0 Core ontology

In addition to these core classes, NIF provides extensions for various, frequent types of linguistic annotations, e.g., words, phrases, sentences, paragraphs as subclasses of strings and typical annotations for these, e.g., parts of speech, named entity annotations, etc. NIF does, however, not provide generic data structures for linguistic annotation.

### Web Annotation

Web Annotation (https://www.w3.org/TR/annotation-model/) provides a RDF-based approach for standoff annotation of web documents, with JSON-LD as its designated serialization. The Web Annotation Data Model provides specifications for the RDF-based annotation of digital resources and the lossless exchange and (re-)usability of such annotations across different media formats, and for all kinds of

annotations. Aside from plain labels, this includes structured elements which may provide, for example, machine readable representations for a particular textual label, e.g., by providing a link with an external ontology. Accordingly, the data model and the vocabulary cover a broad band-width of use cases beyond a plain label-ing mechanism. Instead, annotations are understood as structured objects. The Web Annotation Model provides fully reified representation of annotated elements and annotations assigned to it as summarized for the fragment of the data model in Figure 9.

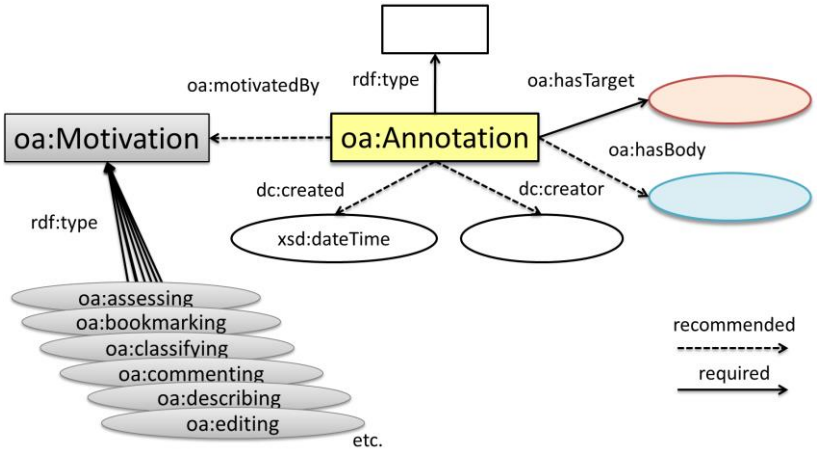



Fig. 9: oa:Annotation and its context in Web Annotation

Aside from string (and other) URIs for targets of annotation, Web Annotation provides various Selectors that define access protocols for elements in a document to be annotated. The TextQuoteSelector matches against every occurrence of a particular string, as illustrated for Named Entity annotations in Figure 10.

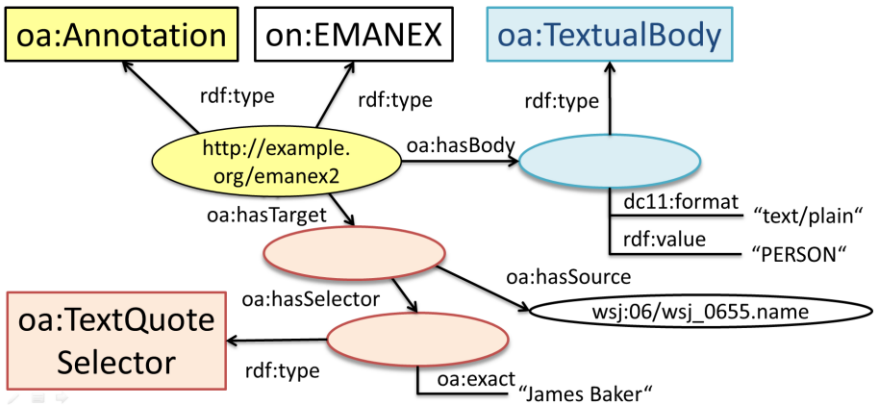



Fig. 10: Annotating all instances of "James Baker" in a Web Annotation document with the named entity type (EMANEX) "PERSON"

Web Annotation is highly generic and applicable to annotate any web content in a LOD-compliant fashion. In comparison with NIF, it has the advantage of being W3C standardized, yet, a downside is that it is overly verbose.

### 3.1.2 Linguistic Annotation Structures

#### CoNLL TSV and related one-word-per-line formats

Since 1999, the Conference on Natural Language Learning (CoNLL, http://www.conll.org/) established a highly successful series of shared tasks in NLP. CoNLL formats are characterized by the use of one word per line, one tab-separated column per annotation layer and an empty line to separate sentences. Subsequently, the shared task formats evolved into a widely used community standard for most forms of linguistic annotations

Here, every word is written in one line, with a series of tab-separated columns holding different annotations; one column contains the surface form of the word. Sentences are separated by an empty line, comments are marked by #. Along with word-level annotations, CoNLL formats support the annotation of spans, illustrated here for named entity annotation using the IOBES scheme, i.e., B-X marking the beginning of the annotation X, E-X its end, I-X intermediate elements, S-X a single-word annotation, and O the absence of an annotation.

```
# WORD      POS   NER         PARSE                  SRL         SRL-ARG
James       NNP   B-PERSON    (TOP (S (NP-SBJ *      _           ARG0
Baker       NNP   E-PERSON    *)                     _           ARG0
Told        VBD   O           (VP *                  tell.v.01   rel
Reporters   NNS   O           (NP *)                 _           ARG2
Friday      NNP   S-DATE      (NP-TMP *)             _           ARGM-TMP
:           :     O           *                      _           _
```

Fig. 11: Integrated CoNLL representation of POS, NER, syntax and PropBank annotations

As shown in Figure 11, word- and span-level annotations can be performed in an intuitive and extensible way with one column per annotation type. A key advantage is that this representation can be easily extended, and easily merged with additional columns. As an example, the conventional way to represent semantic role annotations in CoNLL is to add one column for the predicate as well as another column for every predicate that identifies its arguments. In the fourth column of our example, semantic predicates are identified and marked by a sense identifier. For every predicate instance, its arguments (ARGi with numerical index i for core arguments and ARGM arguments for various modifiers) are represented in a separate column, indicating whether a word occurs in (the span of) a frame argument and in which role.

Closely related to the CoNLL TSV format family are one-word-per-line tabular formats as employed by popular tools in corpus linguistics and digital lexicography, most notably SketchEngine (https://www.sketchengine.eu) and the Corpus Work Bench (http://cwb.sourceforge.net).

## CoNLL-RDF

CoNLL-RDF (https://github.com/acoli-repo/conll-rdf, Chiarcos and Fäth, 2017) is a vocabulary and a converter suite that aims to provide a technological bridge between RDF-based exchange and representation formalisms (such as NIF) and conventional NLP formats (such as CoNLL). Based on a fragment of NIF, CoNLL-RDF provides a semantically shallow and isomorphic reconstruction of TSV formats in RDF and thus represents a technological bridge between the most popular format family in NLP and LLOD technologies.

The listing below shows a CoNLL fragment in the CoNLL-U dialect, the CoNLL format used by the Universal Dependencies (UD) initiative (http://universaldependencies.org/). ID is the number of the word in the sentence, WORD is the form of the word, LEMMA its lemma, UPOS its UD part-of speech tag, POS its original part-of-speech tag, FEATS its morphosyntactic features, HEAD the ID of its parent word in dependency annotation (or 0 for the root), and EDGE the label of its dependency relation. The final columns DEPS and MISC are not used for this example.

```
#ID   WORD       LEMMA    UPOS   POS    FEATS          HEAD  EDGE
1     James      James    PROPN  NNP    Number=Sing    2     name
2     Baker      Baker    PROPN  NNP    Number=Sing    3     nsubj
...
```

The CoNLL-RDF conversion transforms every non-empty, non-comment line to a nif:Word and creates a URI based on the ID column, resp., the number of preceding sentences and words. Every column is mapped to a datatype property, except for the HEAD and SRL-ARGs columns (see above) which are resolved to point to other URIs. Except for nif:Sentence being the conll:HEAD of a nif:Word without syntactic dependency annotation (e.g., the root), these URIs designate nif:Words. Words are connected with nif:nextWord, sentences with nif:nextSentence.

```
:s1_1 a nif:Word;
conll:ID "1"; conll:WORD "James"; conll:LEMMA "James";
              conll:UPOS "PROPN"; conll:POS "NNP";
              conll:FEATS "Number=Sing"; conll:HEAD :s1_2;
              conll:EDGE "name"; nif:nextWord :s1_2.
```

To provide a generic conversion of CoNLL data, CoNLL-RDF expects column labels to be provided at conversion time. For each column, an RDF property is generated using the user-provided label as local name in the conll namespace. As these properties are provided by the user, they lack any alignment to existing RDF/OWL vocabularies. It is in this sense that CoNLL-RDF is shallow as properties are specific for a certain CoNLL format and lack interoperability with other vocabularies.

CoNLL-RDF comes with a Java library for parsing TSV formats into this representation, performing graph transformations with iterative sequences of SPARQL Updates, providing different visualizations and the possibility to export CoNLL-TSV representations for the conll: properties selected by the user as column labels.

### Linguistic Annotation Framework (LAF)

In the context to the ISO TC37/SC4 committee “Language resource management” (https://www.iso.org/committee/297592.html) a series of standards have been dedicated to linguistic annotation. ISO 24612:2012 specifies a linguistic annotation framework (LAF) for representing linguistic annotations of language data such as corpora, speech signal and video. LAF can be considered as a kind of umbrella for other more specific standards, dedicated to morphological annotation (MAF), syntactic annotation (SynAF) and semantic annotation (SemAF), which are all briefly described in the next sections of this deliverable. LAF (and all the associated standards mentioned above) describes an abstract data model, as the main contribution of the normative part of the standard and an XML serialization, as part of the informative section of the standard. It is important to note that the normative part of ISO standards is not available under an open license and can be obtained by paying a fee. But details of the normative parts of LAF have been described in papers, like for example (Ide et al., 2014).

The main motivation for developing LAF (and the associated standards in the ISO TC37/SC4 framework) was to develop a model that can serve a generic representation of what linguistic annotation are and should cover, complementing thus all the linguistic annotation schemes that were used with a formal background.

Later specifications of the LAF approach lead to the GrAF XML pivot format, as linguistic objects can be best considered a representing a graph. But the serialization of this model was still proposed in XML, while the already well developed native graph serializations of RDF have not been proposed. However, the POWLA data model (Sect. 3.1.2.5) does provide a formalization of an equivalent data model in OWL and may be used for the purpose. POWLA is the OWL2/DL reconstruction of the data model of the PAULA XML standoff format that has been developed since 2004 on the basis of early drafts of the LAF and is thus a sibling format to GrAF XML.

LAF and the other ISO TC37/SC4 standards (or specifications) suggest the use of so-called stand-off annotations, so that the documents to be annotated are not undergoing any changes. Relations to the (raw) document are ensured by indices marking positional anchors in the documents to be annotated, pointing to the beginning and the end of elements of the to be annotated in the stand-off annotation set.

An additional important point of the modelling proposed by LAF and related standards is the fact that no vocabulary (“tagsets”) is defined. Rather the model rely on existing vocabulary. In the case of the ISO TC 37/SC4 models the main vocabulary used was ISOcat (cf. Sect. 4.1.1).

Domain-specific adaptations of LAF include

- Morpho-Syntactic Annotation Framework (MAF)

  The Morpho-Syntactic Annotation Framework (MAF), as a kind of sub-project to LAF (see above), was also resulting from work done within the ISO committee TC37/SC4 (http://www.tc37sc4.org). It primary scope is the representation of morpho-syntactic information to be encoded (serialized) in XML. Like LAF (and LMF), it is a generic model. It concerns the annotation of tokens (begin and end of annotation unit are tokens). MAF suggest the use of feature structures to represent the possibly complex morpho-syntactic information to be associated with a token.

  As for LAF, although the model aims at representing a graph (acyclic), the favored serialization is the generation of an XML document (the set of stand-off annotations and the positional anchors linking to the original document). A description of the model is given in (Clément et al., 2005).

- Syntactic Annotation Framework (SynAF)

  Similar to MAF, the Syntactic Annotation Framework (SynAF, with ISO number 24615) proposes a generic model for representing syntactic information. The model considers two types of syntactic information: constituency and dependency. Constituency is represented by nodes and dependency by edges between such nodes. SynAF proposes thus an integrated view on those two types of annotation, within the context of a graph representation. No serialization is proposed in the informative part of the document, as it would be very similar to the serialization proposed in the LAF standard. As for LAF and MAF, a description of the normative part of the standard can be found in a conference paper (Declerck, 2006).

- Semantic Annotation Framework (SemAF)

  A group within the ISO committee TC37/SC4 proposed similar standardisation initiatives concerning semantic information. A series of ISO standards and specifications resulted from this initiative, dealing with “Time and events”, “Dialogue acts”, “Semantic roles”, “Principles of semantic annotation”, “Spatial information”, “Reference annotation framework”, and “Quantification”. Prêt-à-LLOD will study those standards, as far as open documents are available.

### POWLA

POWLA (http://purl.org/powla, Chiarcos, 2012) is an OWL2/DL vocabulary that defines explicit generic linguistic data structures, which can be used in combination with CoNLL-RDF, Web Annotation or NIF to formalize any kind of linguistic annotation. POWLA is grounded in the Linguistic Annotation Framework, and thus capable to represent any kind of text-oriented annotation. This sets it apart from task-specific linguistic data structures provided by NIF or the Web Annotation vocabulary that lacks explicit terminology for linguistic annotations.

Units of annotation are formalized as powla:Node. By means of the property powla:hasParent (and its inverse powla:hasChild), the hierarchical composition of nodes can be expressed, e.g., for syntax trees or discourse structure. The property powla:next (and its inverse powla:previous) is used to express the sequential order of two adjacent nodes. It is recommended that powla:next is used to link nodes which have the same parent, as this facilitates navigation in tree structures. POWLA does not provide its own mechanism for document linking, but can be applied in combination with NIF or Web Annotation, e.g., a NIF URI (representing, for example, a nif:Word) can be linked via powla:hasParent with a powla:Node that represents a syntactic phrase. Aside from Node, POWLA provides reified relations and data structures for annotated corpora, see Fig. 12.

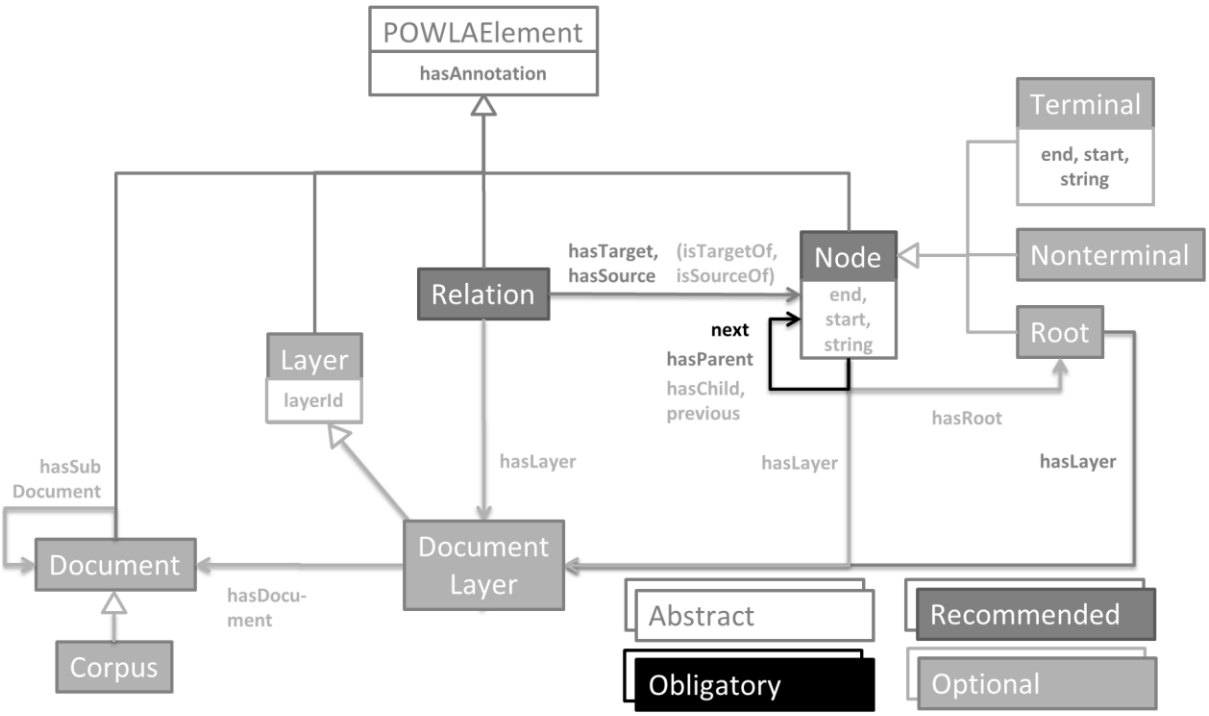



Fig. 12: POWLA data model, obligatory, abstract, recommended and optional properties shown in different shades of grey.

In the context of the Prêt-à-LLOD project, POWLA data structures are used to define an extension of CoNLL-RDF for the processing of tree structures.

## 3.2 New and Emerging Models and Standards

### 3.2.1 Current Shortcomings and Desiderata

The aforementioned vocabularies allow us to anchor linguistic annotations to the elementary units of primary data (XML elements, string URIs, Web Annotation selectors). With respect to linguistic annotations in a linked data context, we identify the following requirements:

- NIF and Web Annotation are relatively verbose, whereas the language resource community prefers simpler representations, either based on (multi-rooted) trees (JSON, XML), or tables (CoNLL/TSV). In order for existing language technology to benefit from Linked Data, it is necessary to provide provide interfaces between pre-RDF and RDF technology.

- CoNLL-RDF provides such a technology, in that it allows to read CoNLL/TSV data, to represent it in RDF, and to export to CoNLL/TSV, again. However, CoNLL-RDF is limited to word- and span-level annotations, whereas extensions to represent syntactic and text-structural trees (additional XML markup in SketchEngine/Corpus Workbench; conventions for encoding syntax trees as a word-level annotation) did not have native support by the CoNLL-RDF vocabulary, resp. converters. Below, we describe the application of the POWLA vocabulary for this purpose. Both CoNLL-RDF and CoNLL-RDF with tree extensions represent extensible vocabularies. In order to facilitate interoperability between different TSV dialects and their respective labels, we will additionally develop an ontology that provides a mapping of conll datatype properties to different CoNLL/TSV dialects.

- TEI/XML is relatively widely used for language resources, but it currently lacks concrete definitions and consistent examples for anchoring RDF annotations in a TEI document. Using Web Annotation, this can be implemented as a W3C-compliant standoff solution by means of selectors, e.g., the XPathSelector, and this functionality is currently provided by the Recogito tool (http://commons.pelagios.org/2018/03/you-can-now-do-everything-in-recogito). Yet, no such specifications for inline XML annotation do exist. For this aspect, the Prêt-à-LLOD project provides a systematic overview over the different possibilities and contributes to the ongoing discussion within the TEI community.

### 3.2.2 CoNLL-RDF Tree Extensions

Beyond CoNLL/TSV files, the CoNLL-RDF converters can be applied to related formats as used by SketchEngine and CorpusWorkbench. These extend CoNLL/TSV by introducing XML markup elements between individual words/lines in order to express a tree structure. So far, this information is, however, omitted by the converter as CoNLL-RDF does not provide vocabulary conventions for representing syntactic (or other) trees. Here, we define the application of POWLA for this purpose and will provide a converter for CoNLL-RDF with POWLA extensions.

A related problem are conventions for representing syntactic (and other) trees as word-level annotations in CoNLL/TSV (e.g., CoNLL-05, CoNLL-11, CoNLL-12): As illustrated in column 4 in Figure 11 above, this involves a modified bracketing notation, originally introduced by the Penn Treebank (https://catalog.ldc.upenn.edu/docs/LDC95T7/cl93.html): Round opening and closing bracket mark the beginning and the end of a phrase, respectively, a label following an opening bracket represents the phrase annotation, any non-parenthetical string content after the label represents the primary data. For representation in CoNLL, the primary data is substituted by the place-holder *, every word is thus annotated with the bracket string that precedes the corresponding * (since the last word) and with the bracket string that follows * (except for bracketing information that applies to the next word). Within CoNLL-RDF, these strings were preserved as string values of a datatype property (e.g., conll:PARSE, if the column name PARSE is provided). Using SPARQL Update, these strings can be evaluated and be used to extrapolate formal data structures, but this requires recursions and is non-performant. Instead, we provide a direct conversion routine, and define here the vocabulary that this converter will produce.

For every opening bracket in the bracketing notation, we produce a powla:Node, with a powla:hasParent relation linking it with its parent, and a powla:next property linking it with its following sibling. Label information is currently provided with the rdf:value property, but may be replaced in the published extension by a property from the conll namespace. It is recommended that powla:Nodes receive absolute URIs based on the number of preceding nodes in their original CoNLL column. An alternative strategy (non-recommended) is to derive phrase URIs from the URIs of the words they cover, so that co-extensional phrases collapse into a single URI. This corresponds to the function of NIF string URIs, which also conflate independent annotations of the same string, the offset-based URI minting strategy avoids this conflation. By application of POWLA semantics, nif:Words in the CoNLL-RDF vocabulary are cast as powla:Words, and connected with its siblings by means of the nif:nextWord property.

For the XML extension, a similar conversion is provided, and the same vocabulary elements are being used. Here, rdf:value provides the element name. Additional vocabulary elements are required for XML attributes: We introduce the namespace http://purl.org/acoli/conll-rdf/xml# and preserve the local name of the attribute name as the name of a datatype property.

Additional vocabulary elements are novel classes: In the bracketing notation, every powla:Node will be given the additional type conll:$COL with $COL being the user-provided column name (say, PARSE). In the XML notation, every powla:Node will be given the type conll:XML_DATA. It is thus possible to recover the original format, if unique phrase URIs are being used.

### 3.2.3 Ontologizing CoNLL-RDF

CoNLL-RDF is an extensible vocabulary, as properties are dynamically generated at conversion time, e.g., from user-provided column labels or the names of XML attributes encountered in the data. It is thus not possible to provide an exhaustive ontology for CoNLL-RDF. Nevertheless, we will provide an OWL2 ontology that provides top-level data structures and a mapping from columns to labels, resp., conll: properties for all TSV formats used in CoNLL Shared tasks in the last 20 years. This ontology will be published as part of the CoNLL-RDF repository and will facilitate the re-usability of CoNLL-RDF data as well as CoNLL-RDF with POWLA extensions.

### 3.2.4 Technological Bridges between TEI/XML and LOD

A specific challenge at the intersection of Linguistic Linked Open Data and the language resource and Digital Humanities communities is the interoperability and the integration of the dominant vocabularies in either field, here for the case of TEI/XML and RDF.
It is possible to extract RDF from TEI using a customized converter and a limited set of phenomena. As for *generic* approaches to bring together TEI/XML and RDF technology, these draw from different motivations, with different solutions. We distinguish three goals:

1. to assert RDF statements about TEI/XML documents,
2. to infuse RDF triples into TEI-compliant inline XML, or
3. to develop TEI/XML (and TEI-generated web documents) into a publication form for (L)LOD.

For the first goal, we recommend a W3C-compliant standoff approach using WebAnnotation/JSON-LD to annotate a TEI document. At the moment, this is the only possibility to link TEI and RDF content in a way that is both TEI-compliant and W3C-recommended. A TEI-compliant alternative is the use of the <xenoData> element that allows to embed, e.g., RDF/XML, JSON-LD or other RDF serializations directly in the header of a TEI document. We can recommend the latter approach only for document-level metadata, as <xenoData> should not contain actual data nor its annotations.

As for the second and third goal, these can be achieved by either

1. (ab)using existing vocabulary elements of the TEI to represent full-fledged RDF triples, or
2. extending TEI/XML with RDFa (https://www.w3.org/TR/rdfa-core/).

Several possibilities for the first approach have been suggested, e.g., using the elements <relation>, <graph>, <link>, <fs> etc. All of these are problematic insofar as these elements are semantically ambiguous between their own, pre-RDF semantics, and an RDF interpretation, and that it is not clear how to map the structure of RDF triples to the respective attributes and child elements. The TEI P5 documentation provides two such examples for <relation> (using *different* attributes), but note that <relation> is (by its parent element) structurally limited to named entities, and not applicable to cross-references between, say texts.

The second approach is the extension of TEI/XML with RDFa attributes. This is a valid TEI customization, it has been prototypically implemented by Prêt-à-LLOD contributors as part of earlier research, and at the moment, this represents the only W3C-compliant way that allows to convey LOD information directly in inline TEI/XML documents, but this is not officially endorsed by the TEI, yet.

Within the project, we will continue and intensify our discussion with the TEI community to help arriving at an official endorsement or and/or a selection among either of these options, cf. https://github.com/TEIC/TEI/issues/311, https://github.com/TEIC/TEI/issues/1860.

# 4. Linguistic Data Categories and Metadata

In this section, we provide a survey over existing resources for linguistic terminology and metadata specifications. This includes

- Linguistic data categories, i.e., abbreviations, tags and concepts for expressing grammatical categories or features in linguistic annotations and lexical-conceptual resources, e.g., with respect to inflectional morphology, agreement and syntactic constructions,
- Vocabularies for language resource metadata, i.e., type and composition of a language resource,
- Metadata for language technology web services, i.e., workflow descriptions, and
- Provenance of linguistic annotations, e.g., as produced by language technology web services.

As for the first two aspects, numerous metadata aggregators with their own specifications exist, as well as several approaches to provide machine-readable metadata. We limit our discussion to specifications developed by or for a broader community and/or beyond a single domain or purpose. We thus exclude resource-specific vocabularies such as those provided by the Universal Dependencies (http://universaldependencies.org), Unimorph (http://unimorph.github.io), or Multext-East (http://nl.ijs.si/ME) that provide specifications for their respective data releases. These vocabularies and resources are, however, subject to metadata and terminology repositories adopting the conventions described below, in particular, OLiA and META-SHARE OWL (resp., META-SHARE OWL v2).

## 4.1 Existing Terminology Repositories and Metadata Specifications

### 4.1.1 Linguistic Data Categories

#### ISO TC37 Data Category Registry (ISOCat)

In the context of the ISO Technical Committee 37 on Terminology and other language and content resources, a metadata registry known as the Data Category Registry (DCR, resp., ISOCat, cf. https://terms.tdwg.org/wiki/ISOcat) was developed along with an associated standard (ISO DIS 12620) for the representation of data categories. The aim of this was to develop a common set of data categories that would provide enough detail such that a domain ontology could be constructed in a “bottom-up” manner from the set of categories contained within the registry.

Data categories were standardized by means of the DCR, an XML schema for the representation of data categories. Each data category record had two main sections: an administration section, which contained key information about the category related to its version, origin and most importantly whether it has been accepted, and a description section consisting of one or more language sections. The description section contains (multilingual)

descriptions of the category including its name, definition, examples as well as the formal definition of the category. The formal definition divided categories into so-called 'simple' and 'complex' categories. Simple categories contain no values, and as such can be seen as equivalent to individuals in OWL ontologies. Complex categories in turn are further divided into three categories: 'open', 'closed' and 'constrained'. Open categories can contain user-defined values and are suitable for extension or for open categories (such as lemmas, or glosses). Closed categories can only take a fixed set of values and are intended for boolean values or for small lists of values. Finally, constrained categories can be limited by e.g. a regular expression, being suitable for an open set of values that follow a certain pattern, such as language tags.

ISOcat adopted an open approach in that any expert can contribute their own data categories with the result that these can be shared with any other user. The work has been thus structured around the thematic domain groups of ISO TC 37. In principle, each of these groups was supposed to manage their individual areas such that when an individual proposes a new category, it would be contributed to one of these TDGs. The approval process was then intended to take a number of steps possibly involving the appointment of extra external experts and either marking it as a duplicate of an existing category, suggesting a hand-off to another TDG, or accepting the category, by which it would be given a unique identifier. The identifier was a number sequentially allocated to each category which could be easily embedded and referenced from an XML or RDF document. For this case the namespace URL http://www.isocat.org/ns/dcr was introduced. The usage of these URLs is not recommended as they do not resolve anymore, as ISOcat development is stalled since 2010.

Successor solutions to ISOcat are being developed with the CLARIN Concept Registry (CCR, https://www.clarin.eu/ccr) and TermWeb (http://demo.termweb.se/termweb/app), but the actual usage, the division of labour, and the interoperability between both portals is yet to be clarified. The CLARIN CCR supports concept URIs and provides a download facility, TermWeb seems to support neither. Both systems use tool-specific formats.

### LexInfo 2.0

LexInfo was designed as an ontology for "associat[ing] linguistic information with respect to any level of linguistic description and expressivity to elements in an ontology" (Cimiano et al. 2011). In the context of Linguistic Linked Open Data, LexInfo is thus the representative vocabulary for linguistic data categories for lexical-conceptual resources. LexInfo predates the Ontolex-lemon model, but was re-designed in parallel with the definition of Ontolex-lemon to become an ontology of linguistic categories with the goal of making Ontolex-lemon itself agnostic of any linguistic category system to support reuse of different linguistic category systems and ontologies in combination with it. For the first release of the LexInfo ontology, a version of the Lexical Markup Framework (Francopoulo et al., 2006) in RDF was used. Version 2.0 was updated to use the Ontolex-lemon model, and many of the functions of LexInfo described originally by Cimiano et al. (2011) are now part of Ontolex-lemon. In terms of its definitions, LexInfo remains to be largely based on LMF, and thus, the definitions developed in the context of ISOcat.

By now, LexInfo has been extended with many extra features, leading it to be one of the most widely-used vocabularies on the Linguistic Linked Open Data Cloud. In particular, LexInfo introduces the following:

- A fixed and axiomatized set of linguistic categories, covering areas such as part of speech, tense, number, animacy, degree, mood, term types (e.g., abbreviation), frequency, register, etc. These categories are partially derived from ISOcat, but with stronger axiomatization (although not as strong as OLiA, covered in the next section).

- Subclasses of Ontolex-lemon's LexicalEntry are introduced by part-of-speech, e.g., Noun, CommonNoun.

- Syntactic frames that are defined by the arguments they require. These are divided first by part-of-speech, then by the set of required arguments, and finally distinguished by their optional (adjunct) arguments. For example, the Transitive class is a subclass of VerbFrame and furthermore is required to have exactly one subject and exactly one directObject. It has a subclass TransitivePP that also admits a prepositional phrase as an adjunct, e.g., "she added salt to the stew". Note that this is distinct from the Ditransitive frame which has a required indirect argument.

- Argument classes and properties are also introduced to enable the axioms for frames to be applied.

- A repertoire to relate senses, lexical entries and forms to each other. For example, translation is defined as a relationship between senses, homonym is a relationship between two entries. pastTenseForm is a relationship between different forms of the same lexical entry.

LexInfo is the reference vocabulary for linguistic categories and features in lexical-conceptual resources in the LLOD community.

### Ontologies of Linguistic Annotation (OLiA)

The Ontologies of Linguistic Annotation (OLiA, http://purl.org/olia, Chiarcos and Sukhareva 2015) provide the reference vocabulary for linguistic annotations in a LLOD context. Unlike LexInfo, OLiA is not a monolithic vocabulary, but instead, provides a modular architecture that uses a central `reference model' to link annotation- or resource-specific terminology (e.g., tag sets) with community-maintained terminology repositories (e.g., ISOcat). With the looming demise of ISOcat and other community-maintained terminology repositories (e.g., GOLD, cf. Langendoen, in press), OLiA has become a terminology resource in its own right and now serves as a central hub for annotation terminology in the LLOD cloud.
OLiA consists of a set of modular OWL2/DL ontologies that formalize the mapping between annotations, a 'Reference Model' and existing terminology repositories ('External Reference Models'):

- The OLiA Reference Model specifies the common terminology that different annotation schemes can refer to. It is based on existing repositories of annotation

terminology and extended for the annotation schemes that it was applied to.

- Multiple OLiA Annotation Models formalize annotation schemes and tagsets. annotation models are based on the original documentation, so that they provide an interpretation-independent representation of the annotation scheme.

- For every annotation model, a Linking Model defines subclass-relationships between concepts/properties in the respective annotation model and the reference model, so that annotation model concepts become interpretable in terms of the reference model. In a similar way, the OLiA Reference Model has also been linked with external reference models such as ISOcat and GOLD.

For two concurrent morphosyntactic annotations of the Brown corpus (within the Penn Treebank, resp., the Susanne Corpus), this is illustrated in Figure 13.

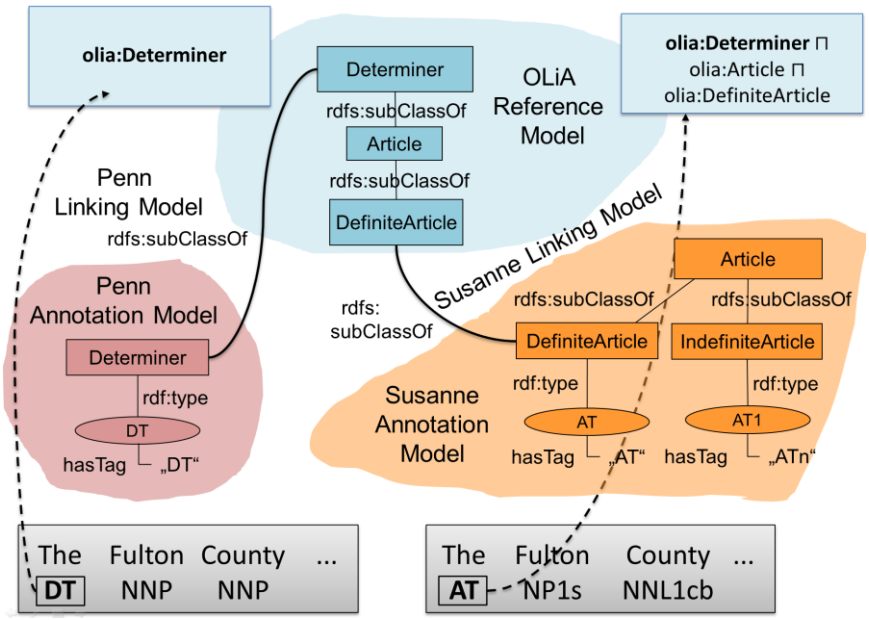


Fig. 13: Interpreting conflicting annotations against OLiA

By linking OLiA reference model concepts with external reference terminologies, a similar interpretation according to ISOcat, GOLD, etc. becomes feasible. Unlike conventional mapping approaches (e.g., in http://universaldependencies.org), the original annotation is, however, left untouched. This approach is thus lossless.

### 4.1.2 Language Resource Metadata

The linked data mechanisms allow metadata of language resources (LRs) to be represented, i.e. , the data that describe the resources themselves (e.g. , typology, languages contained, size of the data, provenance, etc.). Definition of metadata of LRs enables their cataloguing and supports their automated discovery, share and reuse. In this section we give an overview of the most commonly used models to document general metadata of datasets as linked data, that is DC-Terms and DCAT. We will also mention META-SHARE OWL, the most complete vocabulary for describing metadata of LRs as linked data available today.

### DC-Terms

"DC" stands for "Dublin Core", whereas we are not talking about the capital of Ireland but about a city in Ohio, in which in 1995 a first workshop discussing the development of a generic set of metadata for a wide range of resources took place. And as the Web was further growing and with the development of a specific model for describing resources on the Web, the Resource Description Framework (RDF), the Dublin Core vocabulary expanded to a very influential set of metadata that is being for example used extensively used in the Linked Data framework. The whole range of activities of the Dublin Core initiative is available at https://www.dublincore.org. The Dublin Core vocabulary has been standardized, for example in the context of ISO (the most recent specification under the number 15836-1:2017, see also https://www.dublincore.org/collaborations/iso/).

While the original Dublin Core vocabulary was a simple one, comprising 15 generic elements (like contributor, date, publisher, subject, title, etc.) some extensions have been implemented, concerning for example also provenance and copyrights issues. Both the original set and the extensions are now known as DCMI Metadata Terms (DCMI standing for Dublin Core Metadata Initiative) as a single set of terms using the RDF data model, which can be accessed by the URL: http://purl.org/dc/terms, where the users can find all the terms defined and supported by DC in a machine readable format). OntoLex-Lemon is making use of this standardized vocabulary, and is therefore interoperable with all other Web resources that are using the DC terms.

### Data Catalog Vocabulary (DCAT)

The Data Catalog Vocabulary (DCAT) is an RDF vocabulary designed to facilitate interoperability between data catalogs published on the Web (see https://www.w3.org/TR/vocab-dcat). DCAT has the status of W3C Recommendation since January 2014 (https://www.w3.org/TR/2014/REC-vocab-dcat-20140116) and currently is in its 2nd version, recently released as a W3C Candidate Recommendation (https://www.w3.org/TR/vocab-dcat-2). According to its specification, DCAT is an RDF vocabulary designed to facilitate interoperability between data catalogs published on the Web. In fact, DCAT is intended to increase discoverability of datasets by enabling applications to consume and combine metadata from multiple catalogues, enabling also decentralized publishing of catalogs as well as federated dataset search across sites.

DCAT-based data catalogues are organized into datasets and distributions. A distribution is considered an accessible form of a dataset, for instance a downloadable file, a SPARQL endpoint, an RSS feed or a web service that provides the data. DCAT reuses elements from other vocabularies whenever appropriate, such as foaf:homepage, foaf:Agent, dct:title, etc., and defines their own set of core classes:

- *dcat:Catalog* represents a catalog, i.e., a collection of datasets.
- *dcat:Dataset* represents a dataset in a catalog. A dataset is defined as a "collection of data, published or curated by a single agent, and available for access or download in one or more formats".
- *dcat:Distribution* represents an accessible form of a dataset (a downloadable file or web service, for instance).

- *dcat:CatalogRecord* represents the record that describes a dataset in the catalogue. It is used to capture provenance information about dataset entries in a catalogue, and its use is considered optional.

### META-SHARE and META-SHARE OWL

The models referred to above are useful for representing general metadata of LRs (e.g., *title, license, description*). We will focus, in the following, on META-SHARE OWL (http://purl.org/net/def/metashare), a model aimed at representing information that is characteristic of the LRs (*resource type, modality, number of languages*, etc.).

META-SHARE (http://www.meta-share.eu) is an open, integrated, secure, and interoperable exchange infrastructure where language resources are uploaded, documented, stored and catalogued, aiming to support their discoverability and reuse. In order to support such mechanisms, META-SHARE developed a rich metadata schema (Gavrilidou et al., 2012) that allows to describe aspects of language resources accounting for their whole lifecycle, from their production to their usage. The schema has been implemented as an XML Schema Definition (XSD) and descriptions of specific LRs are available as XML documents.

Am OWL version of such schema was developed later on, in the context of the W3C Linked Data for Language Technologies community group (McCrae et al., 2015). The ontology, called META-SHARE OWL, builds on the XML-based model but re-engineered with interoperability in mind and to maximise compatibility with other vocabularies such as DCAT or the most prominent models in the CLARIN VLO data. META-SHARE-OWL defines many ontology entities for describing language resources but also reuses a number of entities coming from other vocabularies to account for general metadata.

The META-SHARE OWL ontology significantly re-structured the original XML-based model in order to avoid unnecessary or overly verbose nodes in the produced RDF graph. The resulting OWL ontology has 192 classes and 358 properties, which enables a very rich and fine-grained description of metadata of LRs. The core class in the META-SHARE OWL ontology, used to describe the most relevant features of a LR, is:

- *ms:LanguageResource* is the core class in the ontology and represents a language resource and has the following specializations:
  - *ms:Corpus*, which identifies written/text, oral/spoken, multimodal/multimedia corpora in one or several languages.
  - *ms:LexicalConceptualResource*, which represents lexical-conceptual resources, such as terminologies, glossaries, word lists, dictionaries, semantic lexica, ontologies, etc.
  - *ms:LanguageDescription*, which represents resources that describe a language, such as grammars (set of rules that describe a language formally) or language models (containing statistical information).
  - *ms:ToolService*, which represents tools and services for language processing.

Then, a number of properties can be used to describe the particular features of such a resource, such as *ms:LingualityType* (e.g., monolingual, bilingual), *ms:Size*, *ms:CharacterEncoding* (e.g., *ms:UTF-8, ms:ISO-2020-JP*), *ms:ModalityType* (e.g., *ms:writtenLanguage, ms:signLanguage*), etc.

### 4.1.3 Language Technology Service Metadata

As a complement to existing platform solutions such as Apache UIMA or proprietary end-to-end NLP stacks, web technologies allow to integrate and to combine different specialized components developed by independent contributors. Such multi-provider architectures are especially important in the context of the European language technology market where much of the technology is being provided by SMEs rather than a single tech giant. For such architectures, web services are the state-of-the-art approach and a particularly convenient way to provide and to access natural language processing modules, but integration of modules, resources and components depends on common metadata specifications. The metadata of an NLP web service provides all the information needed in order to interact with it. Such metadata includes a description of functionalities offered by a service, pre and postconditions, and specifications of data that is consumed and produced by a service. This information makes it possible to invoke a service successfully, providing it with the data it needs for processing, and enabling interpretation of the results. In this section, we discuss various aspects of metadata specification of NLP web services of two selected state-of-art NLP frameworks for building and executing natural language processing pipelines, CLARIN and the LAPPS Grid. We describe the underlying workflow engines, the data formats and the semantic requirement of metadata specification of each framework.

#### CLARIN Web Service Metadata

WebLicht[20] (Dima et al., 2012) is an environment for building and executing natural language processing pipelines, integrated into the CLARIN[21] infrastructure (Hinrichs and Krauwer, 2014), which provides easy access to a wide range of text processing tools to researchers in the humanities and social sciences. It is built upon Service Oriented Architecture (SOA) principles, which means that processing tools are implemented as web services that are hosted on servers distributed across the web. WebLicht NLP tools are implemented as web services that consume and produce the Text Corpus Format (TCF)[22] data, an XML format designed for use as an internal data exchange format for WebLicht processing tools. The TCF also ensures semantic interoperability among all WebLicht tools and resources by defining a common vocabulary for linguistic concepts described in TCF Schema[23]. Metadata descriptions of WebLicht tools are stored in repositories located at the CLARIN center hosting the service. WebLicht web services are invoked using the REpresentational State Transfer (RESTful) API. Each time a service is added to a workflow, the cumulative output of the workflow is calculated by inspecting the output descriptions in the metadata for each tool in the workflow. A workflow is executed by sequentially invoking each service in the pipeline,

[20] https://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php/Main_Page
[21] https://www.clarin.eu/
[22] https://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php/The_TCF_Format
[23] https://github.com/weblicht/tcf-spec/blob/master/src/main/rnc-schema/textcorpus_5.rnc

passing the output of one service as input to the next. Therefore, service metadata of NLP tools developed in CLARIN framework plays an important role to create, build, process and visualize NLP processing pipelines.

WebLicht web service metadata is based on the Component Metadata Infrastructure (CMDI)[24], an XML-based framework developed within CLARIN that provides a way to describe and reuse metadata components. The CMDI model has close ties to the ISOcat data category registry[25] which provides clear and unambiguous semantics when using metadata. The model can also be easily extended to other widely agreed registries of data categories. There are two supported versions of CLARIN's component metadata framework: CMDI 1.1 and CMDI 1.2[26]. They are not interchangeable. The metadata descriptions for all tools and web services of WebLicht are open in repositories of CLARIN centers[27]. Therefore it allows arbitrary service providers to harvest the descriptions via accepted protocols such as OAI-PMH[28]. Figure 14 shows the life cycle of CMDI metadata.

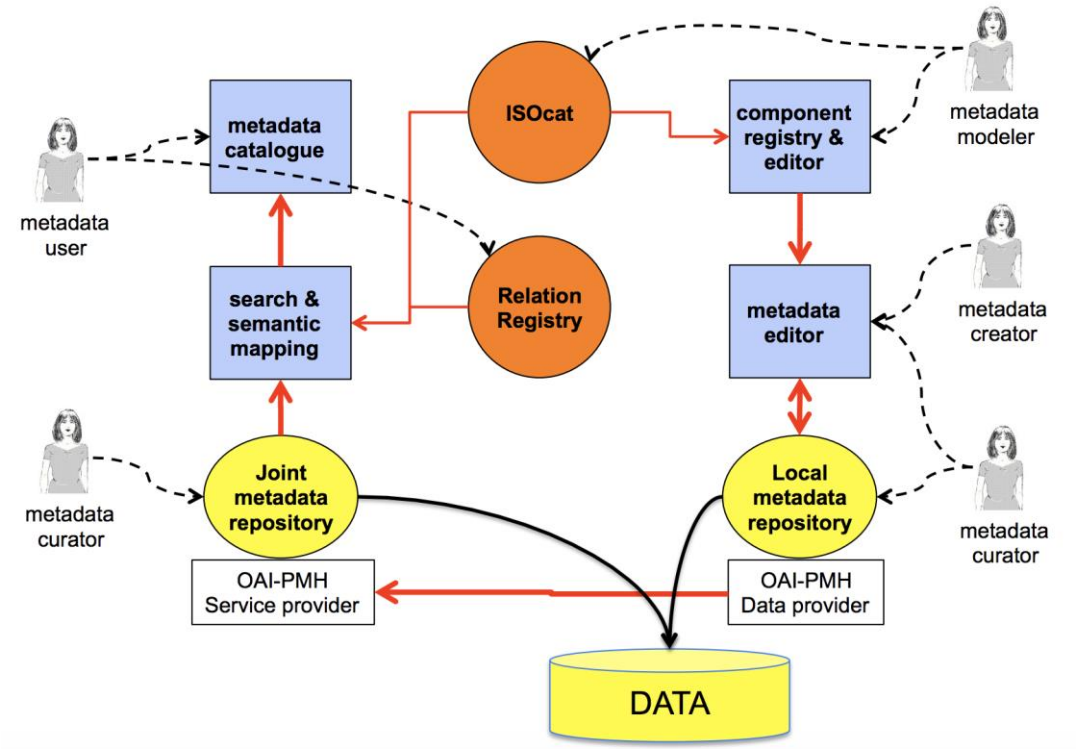


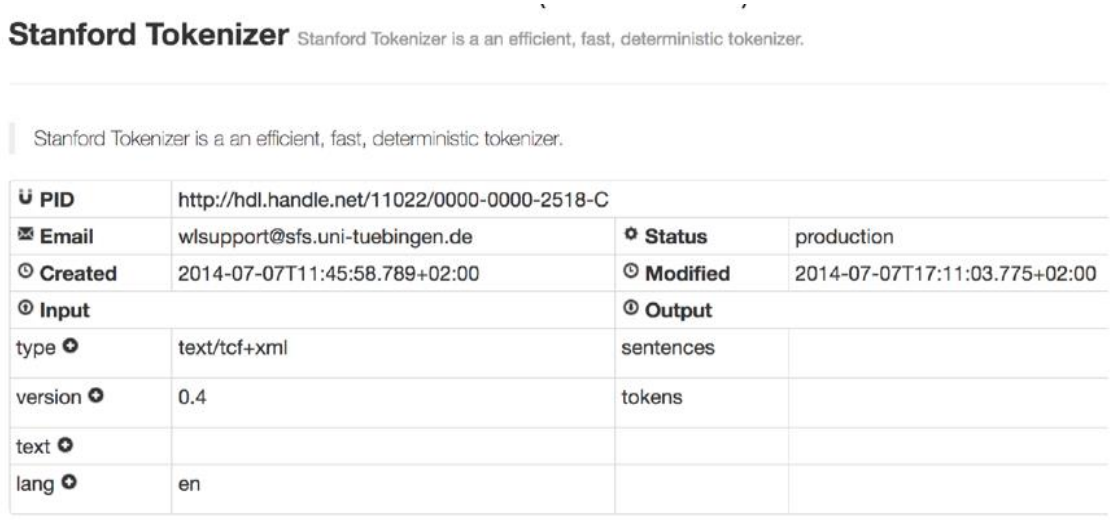

Fig. 14: CMDI metadata format life cycle

The CMDI approach combines architectural freedom when modeling the metadata with powerful exploration and search possibilities over a broad range of language resources. Components are building blocks of information (e.g. name or email) which can be grouped to form profiles (e.g. contact-person). Both the components themselves and the profiles that

[24] http://www.clarin.eu/cmdi
[25] https://media.dwds.de/clarin/userguide/text/concepts_ISOcat.xhtml
[26] https://www.clarin.eu/cmdi1.2-specification
[27] https://centres.clarin.eu/oai_pmh
[28] https://www.openarchives.org/pmh/

are built with them are stored in the CLARIN Component Registry. The CMDI WebLichtWebService[29] profile format provides two types of information:

- The *General Information* contains information about creators, access rights, development status, service description, and PID (a unique ID for a web service).
- The *Orchestration Information* contains information needed to invoke the service, such as input requirements and output description such as expected Input (data type and annotations required to be in the input), output produced (data type and list of annotation layers added), URL, query parameters, etc.



**Stanford Tokenizer** Stanford Tokenizer is a an efficient, fast, deterministic tokenizer.

Stanford Tokenizer is a an efficient, fast, deterministic tokenizer.

| | | | |
|---|---|---|---|
| **PID** | http://hdl.handle.net/11022/0000-0000-2518-C | | |
| **Email** | wlsupport@sfs.uni-tuebingen.de | **Status** | production |
| **Created** | 2014-07-07T11:45:58.789+02:00 | **Modified** | 2014-07-07T17:11:03.775+02:00 |
| **Input** | | **Output** | |
| type | text/tcf+xml | sentences | |
| version | 0.4 | tokens | |
| text | | | |
| lang | en | | |

Fig. 15: An example of service metadata of Stanford Tokenizer of WebLicht.

Figure 15 shows an example of metadata in CMDI format of a Stanford Tokenizer implemented in WebLicht. WebLicht uses the CMDI Orchestration Metadata Editing Tool (COMET)[30], which is a tool for creating, editing, and validating WebLicht service metadata for adding a new service in CLARIN framework. The metadata is then added to the CLARIN repository for subsequent harvesting by WebLicht.

### LAPPS Web Service Metadata

The LAPPS Grid[31] (Ide et al., 2014b) is a framework that provides access to basic NLP processing tools and resources and enables pipelining these tools to create custom NLP applications, as well as access to language resources such as mono- and multilingual corpora and lexicons that support NLP. In the LAPPS Grid, language resources and NLP tools are made available as web services through the Galaxy[32] workflow engine and interface (Giardine et al., 2005), as well as programmatic access through the LAPPS Grid application programming interface API[33]. LAPPS Grid tools consume and produce data in

[29] https://catalog.clarin.eu/ds/ComponentRegistry/#/?_k=96wfwx
[30] http://weblicht.sfs.uni-tuebingen.de/comet/
[31] http://www.lappsgrid.org/
[32] http://galaxyproject.org
[33] http://wiki.lappsgrid.org/Developing.html

the LAPPS Interchange Format (LIF)[34] (Verhagen et al., 2016), a JSON-LD (i.e., RDF) format designed to serve as an internal interchange format for linguistically annotated data. NLP tools are accessed as web services that deliver metadata about the content at a standardized URI and are at present invoked using the Simple Object Access Protocol (SOAP)[35]. Figure 16 illustrates the overall workflow engine of LAPPS.

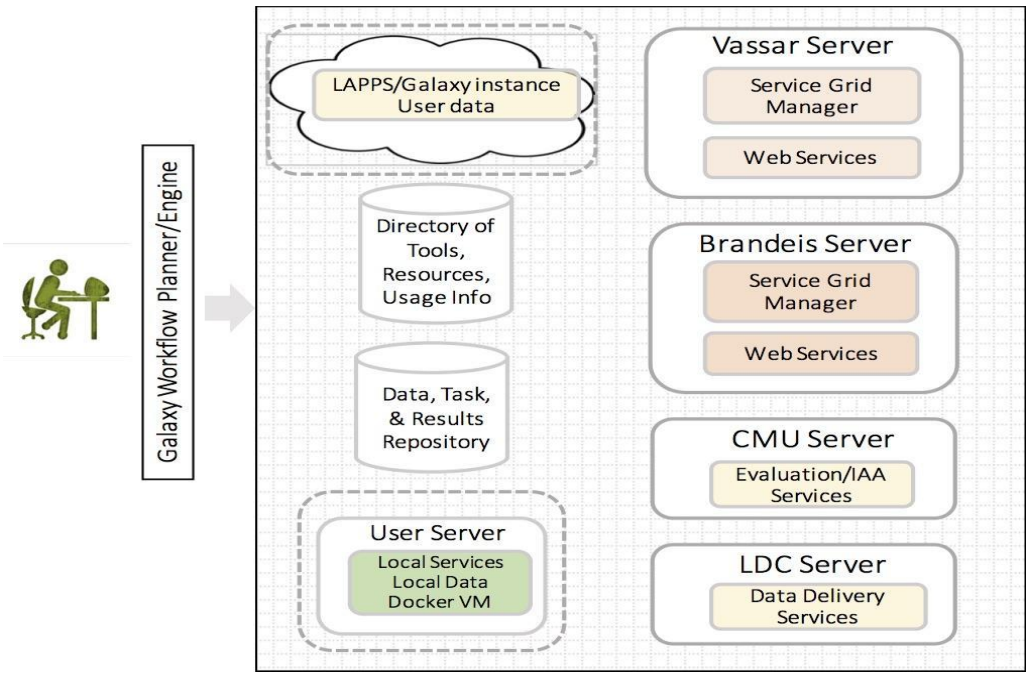



Fig. 16. shows the workflow of LAPPS grid.

For service metadata, the LAPPS Grid uses the Simple Object Access Protocol (SOAP), a messaging protocol for exchanging information via the internet, for invoking web services. In the SOAP protocol, each web service provides its own metadata. The format of a SOAP message is written in the Extensible Markup Language (XML), a simple, flexible text format derived from the Standard Generalized Markup Language (SGML) which is developed by the International Organization for Standardization (ISO 8879:1986). XML Schema describes the structure and contents messages received by and sent by Web services. Figure 17 shows an example Stanford Tokenizer tool of LAPPS metadata format. The repositories of Brandeis[36] and Vassar[37] maintain LAPPS Grid web services and provide service discovery functionalities to users and applications. LAPPS Grid services provide the following metadata information:

- General information about the tool (name, description, vendor, licensing)
- Input requirements (data type, language and encoding, required previous annotations)
- Output produced (data type, language and encoding, output annotations)

[34] https://wiki.lappsgrid.org/interchange/
[35] https://www.w3.org/TR/soap12/
[36] http://api.lappsgrid.org/services/brandeis
[37] http://api.lappsgrid.org/services/vassar

| | |
|---|---|
| Name | org.anc.lapps.stanford.Tokenizer |
| URL | http://vassar.lappsgrid.org/invoker/anc:stanford.tokenizer_2.0.0 |
| Version | 2.0.0 |
| Description | Stanford Tokenizer |
| Vendor | http://www.anc.org |
| Allow | http://vocab.lappsgrid.org/ns/allow#any |

**Requirements**

| | |
|---|---|
| Language | en |
| Formats | http://vocab.lappsgrid.org/ns/media/jsonld#lif |

**Produces**

| | |
|---|---|
| Language | en |
| Formats | http://vocab.lappsgrid.org/ns/media/jsonld#lif |
| Annotations | http://vocab.lappsgrid.org/Token |

Fig. 17: An example of service metadata of Stanford Tokenizer of LAPPS.

Semantic interoperability among services is accomplished via URI references to the LAPPS Grid Web Service Exchange Vocabulary (WSEV) (Ide et al., 2016). The WSEV is prepared in collaboration with ISO TC37 SC4 WG1 in order to ensure full community engagement and input. It provides a single web location where terms relevant for exchange among NLP tools are defined and provides a “sameAs” link to all known web-based definitions that correspond to them. It also defines relations among the terms that can be used when linguistic data is exchanged at LIF data interchange format. The WSEV is intended to be used by a federation of grids currently being formed, including the Kyoto Language Grid, the Language Grid Jakarta Operation Center, the Xinjiang Language Grid, the Language Grid Bangkok Operation Center, LinguaGrid, MetaNET/Panacea, and LAPPS, but is usable by any web service platform.

### 4.1.4 Provenance of Linguistic Annotations

We conclude the description of metadata vocabularies with an excourse about provenance as one particularly important type of linguistic metadata that requires a detailed discussion, albeit it is secondary to the linguistic data it is applied to: Provenance information is metadata describing facts related to the creation process of a resource or entity. In the context of NLP data, provenance information typically consists of the creation time, information about human agents who caused or performed the creation or modification of a datum, and of a description of the pieces of software that were involved in the creation process. In the context of the web of data, the standard vocabulary for this purpose is PROV-O.

### PROV-O: The PROV Ontology

PROV-O (https://www.w3.org/TR/prov-o/) is centered around three major concepts: entities, agents, and activities. Entities are products and similar tangible results of some activity that was caused or influenced by one or more agents (these can be human or non-human agents, such as pieces of software). Also, the creation of an entity regularly involves other entities on which the resulting entity is based.

Thus, PROV-O ontology allows users to model these central concepts for any resource, event or situation for which provenance information is required, and to attach properties to these central entities that give more information. The three central entity types in PROV-O are the following:

- **Entities:** Central descriptive information (such as title or description), information about the type of content contained within the entity, information about other related entities.
- **Agents:** Information about central properties of the agent (it is advisable to reuse the existing FOAF vocabulary here).
- **Activities:** Information about the time of performance of an activity, connections to related or involved entities, connections to involved agents.

The schema of PROV-O is shown in Figure 18 (taken from https://www.w3.org/TR/prov-o/):

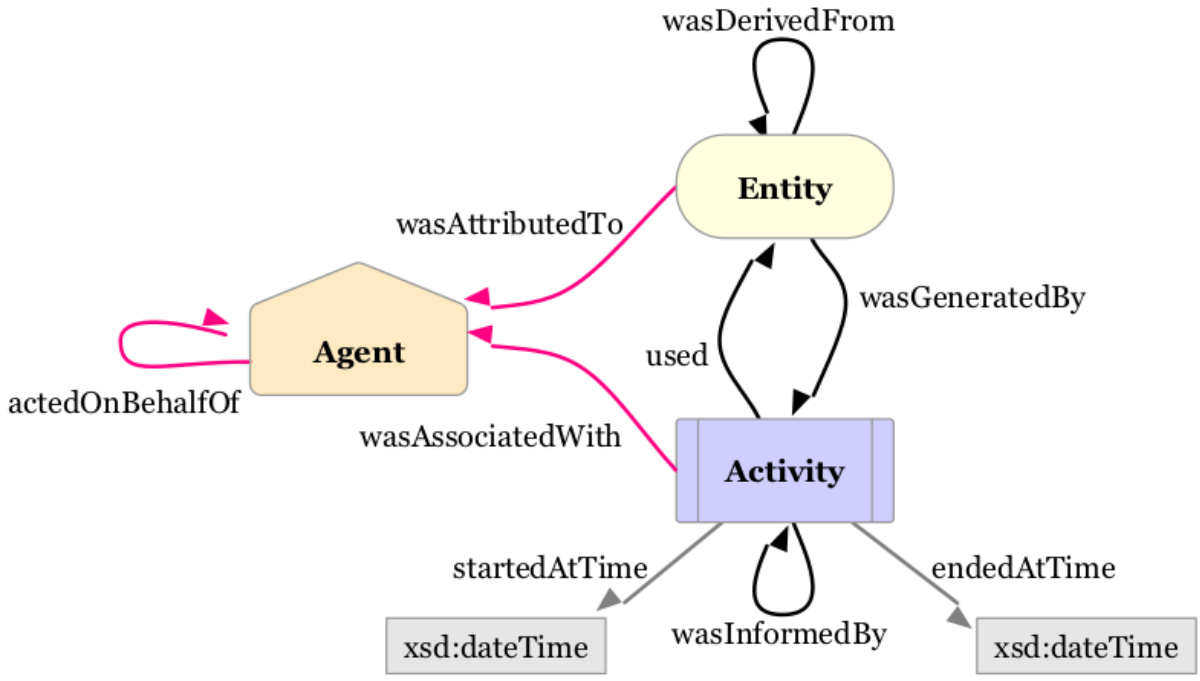



Fig. 18: Schema of PROV-O

The following RDF example shows how to define the provenance of a NIF annotation using PROVO-O. The example shows a token “The” that is related via the property nif:annotation to three annotations corresponding to a dependency relation annotation (ex:Dep12), lemma annotation (ex:Lemma0) as well as POS annotation (ex:Pos0).

```
<http://example.org/tcf2nif/example.txt#char=0,3> a nif:RFC5147String,
        nif:String, nif:Word ;
    nif:anchorOf "The" ;
    nif:annotation ex:Dep12, ex:Lemma0, ex:Pos0 ;
    nif:beginIndex "0"^^xsd:nonNegativeInteger ;
    nif:endIndex "3"^^xsd:nonNegativeInteger ;
    nif:referenceContext <http://example.org/tcf2nif/example.txt#char=0,> ;
    prov:generatedAtTime "2015-07-09T14:01:00"^^xsd:dateTime ;
    prov:wasDerivedFrom <http://example.org/tcf2nif/example.txt#char=0,> ;
    prov:wasGeneratedBy ex:TokenizationActivity .
```

The example shows how to add provenance to the tokenization proper using properties from the PROV-O ontology. Besides, it should be obvious how to add provenance information to the three annotation objects ex:Dep12, ex:Lemma0, and ex:Pos0.

Overall, PROV-O is certainly the standard for representing provenance information and is recommended to be used in combination with NIF and other annotation ontologies to represent the origin of annotations. However, as (Menke et al., 2017) noted, the plain approach to attaching provenance information to each annotation is very inefficient, leading to a proliferation of triples. The authors propose a modular approach to annotation of NIF with PROV-O by attaching annotations to modules rather than single annotations. This is shown to decrease the number of triples needed significantly.

In the context of Prêt-à-LLOD, we found PROV-O to be an adequate solution and plan to apply it along these lines.

## 4.2 New and Emerging Models and Standards

### 4.2.1 Current Shortcomings and Desiderata

We observe the following problems:

- Increasing fragmentation of resources for linguistic data categories

  With respect to linguistic data categories, the current decade has seen a transition from a monolithic, albeit poorly structured resource (ISOcat) to a growing number of highly structured, but domain- or application-specific vocabularies. The latter include the ISOcat successor systems TermWeb (for multilingual terminology) and CLARIN-CCR (for language technology), native LLOD vocabularies such as LexInfo (for lexical-conceptual resources) and OLiA (for linguistic annotations), but also novel vocabularies developed by recent, influential community efforts, e.g., the Universal Dependencies (http://universaldependencies.org, a collection of corpora with cross-linguistically comparable annotations for dependency syntax, parts of speech, and morphosyntactic features) and Unimorph (http://unimorph.github.io, a multilingual collection of morphologically tagged word forms with their respective lemmata). Within Prêt-à-LLOD, we aim to facilitate the interoperability of these vocabularies by linking them with OLiA and thus, with each other.

- Updating existing LLOD vocabularies

  Recent years have seen an immense growth in language resources provided as linked data. META-SHARE OWL and Lexinfo require updates of different extent to reflect these developments.

  As for the META-SHARE OWL ontology, early adopters of such a model (e.g., the OpenMinted and the ReTeLe projects) detected improvement aspects that could help to simplify the model. Most of them were artifacts inherited from the original XML version that are not really necessary in a purely graph-oriented version of the metadata model.

  Likewise, LexInfo requires updates, but at a lower scale, as it is designed to complement Ontolex-lemon (resp., lemon, its predecessor), and LexInfo 2.0 is not yet synchronized with some more recent developments of the vocabulary.

- Lack of interoperability between and within NLP web service architectures

  The LAPPS and CLARIN architectures as described above differ greatly in their interface and metadata specifications.

  - *SOAP vs. REST*: LAPPS uses SOAP message for exposing web service metadata and it suffers from limitations of SOAP architecture. SOAP has no structured way to deal when wrong SOAP messages are exchanged and processed unnoticed. It is very challenging to debug problems in SOAP messages. Therefore, SOAP messaging protocol is losing popularity and RESTful web service is taking its place in the web industry.

  - *System-specific metadata*: The metadata structure, storing, conversion, communication protocol, and the fetching process is well organized in CLARIN infrastructure. However, it is very complicated and expensive to integrate CLARIN web service with other NLP infrastructure. Mapping any web service metadata to CLARIN metadata requires additional information (such as creators, the short and long description of the service, development status, etc.) relevant to CLARIN CMDI framework. To enter this additional information CLARIN provides a CMDI Orchestration Metadata Editing Tool (COMET), which is a tool for creating, editing, and validating WebLicht service metadata.

  - *Annotation compatibility:* The LAPPS Grid web service metadata provides basic information of an NLP tool such as input and output specifications. LAPPS does not attempt to integrate what is commonly referred to as tagsets in metadata specification. The tagsets are used in annotation (e.g., part-of-speech, dependency relations, constituent names, etc.) and it is necessary to incorporate a way to present this information in web service metadata. There have been attempts to map and/or harmonize such values (e.g., OLiA (Chiarcos, 2008)), which have amply demonstrated the difficulties of this kind

of mapping. The lacking of presenting tagsets in metadata specification causes a problem in creating, building, and executing NLP pipelines with taggers and parsers of different tagsets.

### 4.2.2 META-SHARE OWL v2 Ontology

At the time of writing, a new version of the META-SHARE OWL ontology is being developed, being backwards compatible with the initial one, however removing all the artifacts inherited from the XML version that hampers a simplified use of the model. The status of the new version of the ontology can be checked at https://github.com/ld4lt/metashare.

This ongoing effort is primarily led by partners of the European Language Grid project, with the most substantial contribution by Athena (Greece), but receiving inputs and contributions also from Prêt-à-LLOD partners (NUIG, UNIZAR, UPM, DFKI).

It is expected that, as a result, a re-built OWL ontology will be in place that will be able to support the representation of metadata of language resources in a more comprehensive and interoperable way, and will serve as basis for the new Linghub version to be developed in Prêt-à-LLOD.

### 4.2.3 Updates to Lexinfo

One specific challenge is keeping up-to-date with changes related to adapting category systems to new languages and domains. There have been many criticisms of models including LexInfo for not mapping to some languages (Chavula et al., 2014) and LexInfo 2.0 has not been updated to the most recent changes in the OntoLex vocabulary. While LexInfo is editable via GitHub, this interface is too technical to be practical. Instead work is planned on the collaborative development of an interface for defining linguistic categories formally using OWL, without failing into some of the traps of previous attempts such as ISOcat (Schuurman et al., 2015) and it is expected that Prêt-à-LLOD technology will be key to these efforts.

### 4.2.4 Linking Terminology Repositories via OLiA

We aim to counter the increasing fragmentation of linguistic terminology resources by linking them with the OLiA Reference Model as novel External Reference Models (if they define concepts), resp., Annotation Models (if they define values):

- Add CLARIN CCR as an external reference model, i.e., define OLiA reference model classes as subclasses of CLARIN CCR items. This mapping will be guided by the existing ISOcat linking of OLiA, using ISOcat identifiers maintained in CLARIN CCR.
- Add LexInfo 2.0 as an annotation model, i.e., define (selected) LexInfo classes and individuals as subclasses, resp. instances of OLiA reference model classes. This mapping will be guided by the existing ISOcat linking of OLiA, as LexInfo largely builds on ISOcat.
- Provide OLiA annotation models for the Universal Dependencies (UD), resp., their language-specific editions. The mapping of grammatical features will be guided by

the existing linking of MULTEXT-East with OLiA, as MULTEXT-East is partially underlying Interset (Zeman, 2008) which represents the basis for UD feature annotations. The mapping of dependency labels will be guided by the OLiA linking for the Stanford dependencies. A prototype for the mapping of UD v1 specifications has been developed in preparation of the EUROLAN Summer School 2015 and will be updated to UD v2, currently covering more than 100 treebanks in over 70 languages.

- Provide OLiA annotation models for Unimorph, resp., its language-specific editions. For selected languages, a prototypical mapping has been developed in preparation of Prêt-à-LLOD. This will be systematically extended to all (currently 110) Unimorph languages.

With these terminology repositories linked to the OLiA Reference Model, it will become possible to derive mappings between all of them, and between them and earlier terminology repositories such as GOLD, and ISOcat. Note, however, that we do not guarantee 1:1 mappings, but rather 1:*n* and *m*:1 mappings as granularity differences are maintained in OLiA rather than being levelled (as in Universal Dependencies).

### 4.2.5 Web Service Interoperability

Recent efforts to improve interoperability within and between NLP web service architectures include for example, a project to integrate the Language Applications (LAPPS) Grid and CLARIN, a project funded by the Andrew K. Mellon Foundation.[38] The goal was to enable seamless interoperability at both the syntactic and semantic levels among tools available from both the LAPPS Grid and WebLicht (i.e. CLARIN) so that users can mix and match these tools regardless of provenance and without concern for differing I/O requirements. A major task of this integration process was to develop a web service that fetches WebLicht metadata and converts it to the LAPPS metadata format. A shortcoming is that it combines two versions of the CMDI format (CMDI 1.1 and CMDI 1.2), and they are not interchangeable.

In the context of Prêt-à-LLOD, these issues are addressed by bundling NLP functionalities in Docker containers and by developing a designated vocabulary to describe their dependencies and the formation of workflows on that basis. This will be directly grounded in earlier experiences from CLARIN and LAPPS and be developed as part of the Teanga workflow system (WP 3, Task 3.3). For the specific task of annotation interoperability, we plan to refer to the OLiA ontologies and to develop an annotation transformation component on that basis (WP3, Task 3.1).

[38] https://mellon.org/grants/grants-database/grants/brandeis-university/1901-06505/

# 5. Summary

With this report, we provide a survey over representative pre-RDF and RDF-based vocabularies for various interoperable language resources, resp., services that produce corresponding annotations.

We have shown that adequate LLOD vocabularies for all aspects considered do already exist, but that several shortcomings and desiderata can be identified, especially with respect to facilitating interoperability *beyond* resources of a specific type or domain, as these are beyond the scope of pre-RDF vocabularies. With respect to lexical-conceptual resources, linguistic annotation, and linguistic data categories and metadata, we identified directions to be pursued within the Prêt-à-LLOD project to address these gaps and described new and emerging models developed under participation of Prêt-à-LLOD partners. We expect that the Prêt-à-LLOD project will contribute greatly to the development of a mature stack of LLOD vocabularies for interoperable language resources.

# 6. References


Alsina, V. and DeCesaris, J. (1998). Morphological structure and lexicographic definitions: The case of-ful and-like. In: *Proceedings of the 8th EURALEX International Congress, EURALEX 1998*. Liège, Belgium, pp. 545–554.

Baker, C. F., Fillmore, C. J. and Lowe, J. B. (1998). The Berkeley FrameNet project. In *Proceedings of the 17th International Conference on Computational Linguistics, COLING 1998.* Montreal, Quebec, Canada, vol. 1,pp. 86–90.

Berant, J., Dagan, I. and Goldberger, J. (2011). Global learning of typed entailment rules. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, HLT 2011*. Portland, Oregon, vol. 1, pp. 610–619

Bond, F., Vossen, P., McCrae, J.P. and Fellbaum, C. (2016). CILI: The Collaborative Interlingual Index. In *Proceedings of the 8th Global WordNet Conference, GWC 2016*. Bucharest, Romania.

Brown, S., Bonial, C., Obrst, L. and Palmer, M. (2017). The Rich Event Ontology. In *Proceedings of the Events and Stories in the News Workshop*. Vancouver, Canada, pp. 87–97.

Chavula, C. and C. Keet, C. M. (2014). Is lemon sufficient for building multilingual ontologies for Bantu languages? In *Proceedings of the 11th International Workshop on OWL: Experiences and Directions, OWLED 2014*, Riva del Garda, Italy, vol. 14, pp. 61–72.

Chiarcos, C. (2012). POWLA: Modeling linguistic corpora in OWL/DL. In *Proceedings of the Extended Semantic Web Conference, ESWC 2012*, pp. 225–239.

Chiarcos, C. (2008). An ontology of linguistic annotations.LDV Forum , 23(1):1–16.

Chiarcos, C., and Fäth, C. (2017). CoNLL-RDF: Linked corpora done in an NLP-friendly way. In *Language, Data, and Knowledge, LDK 2017.* Galway, Ireland, pp. 74–88.

Chiarcos, C. and Sukhareva, M. (2015). OLiA – Ontologies of Linguistic Annotation. In *Semantic Web*, 6(4), pp. 379–386.

Cimiano, P., Buitelaar, P., McCrae, J., and Sintek, M. (2011). LexInfo: A declarative model for the lexicon-ontology interface. In *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 9(1), pp. 29–51.

Cimiano, P., McCrae, J.P. and Buitelaar, P. (eds., 2016). Lexicon Model for Ontologies. W3C Ontology-Lexica Community Report. URL https://www.w3.org/2016/05/ontolex/

Clément, L. and Villemonte de La Clergerie, É. (2005). MAF: A morphosyntactic annotation framework. In *Proceedings of the 2nd Language & Technology Conference*. Poznań, Poland, pp. 90–94.

Declerck, T. (2006). Synaf: Towards a standard for syntactic annotation. In *Proceedings of*

*the 5th Conference on International Language Resources and Evaluation, LREC 2006*. Genova, Italy.

Declerck, T., Egorova, K. and Schnur, E. (2018). An integrated formal representation for terminological and lexical data included in classification schemes. In *Proceedings of the 11th International Conference on Language Resources and Evaluation, LREC 2018*. Miyazaki, Japan.

Dima, E., Hinrichs, E., Hinrichs, M., Kislev, A., Trippel, T., and Zastrow, T. (2012). Integration of weblicht into the clarin infrastructure. In *Proceedings of the Joint CLARIN-D/DARIAH Workshop at Digital Humanities Conference 2012: Service-oriented Architectures (SOAs) for the Humanities: Solutions and Impacts*, Hamburg, Germany, pp. 17–23.

Fillmore, C. J. (1968). The Case for Case. In *Universals in Linguistic Theory*. Holt, Rinehart and Winston, London, United Kingdom, pp. 1–25.

Francopoulo, G., George, M., Calzolari, N., Monachini, M., Bel, N., Pet, M., Soria, C., et al. (2006). Lexical Markup Framework (LMF). In *Proceedings of the 5th Conference on International Language Resources and Evaluation, LREC 2006*. Genova, Italy.

Gangemi, A., Alam, M., Asprino, L., Presutti, V. and Recupero, D. R. (2016). Framester: A wide coverage linguistic linked data hub. In *European Knowledge Acquisition Workshop*. Springer, Cham, pp. 239–254.

Gavrilidou, M., Labropoulou, P., Desipri, E., Piperidis, S., Papageorgiou, H., Monachini, M., Frontini, F., Declerck, T., Francopoulo, G., Arranz, V., Mapelli, V. (2012). The META-SHARE metadata schema for the description of language resources. In *Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012*. Istanbul, Turkey.

Giardine, B., Riemer, C., Hardison, R. C., Burhans, R., Elnitski, L., Shah, P., Zhang, Y., Blankenberg, D., Albert,I., Taylor, J., Miller, W., Kent, W. J., and Nekrutenko, A. (2005). Galaxy: a platform for interactive large-scale genome analysis. In *Genome Research*, vol. 15(10), pp. 1451–55.

Hinrichs, E. and Krauwer, S. (2014). The CLARIN research infrastructure: Resources and tools for eHumanities scholars. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014*, Reykjavik, Iceland.

Ide, N., Suderman, K. (2014a). The Linguistic Annotation Framework: A standard for annotation interchange and merging. *Language Resources and Evaluation*, vol. 48, pp. 395–418.

Ide, N., Pustejovsky, J., Cieri, C., Nyberg, E., Wang, D., Suderman, K., Verhagen, M., and Wright, J. (2014b). The language applications grid. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014*, Reykjavik, Iceland.

Ide, N., Suderman, K., Verhagen, M., and Pustejovsky, J. (2016). The language applications grid web service exchange vocabulary. In *Revised Selected Papers of*

*the Second International Workshop on Worldwide Language Service Infrastructure WLSI (2015)*. Kyoto, Japan, pp. 18–32.

Kingsbury, P., and Palmer, M. (2003). PropBank: The next level of treebank. In *Proceedings of the 2nd Workshop on Treebanks and Lexical Theories, TLT 2003*. Växjö, Sweden.

Langendoen, T. D. (in press). Whither GOLD? In Pareja-Lora, A., Blume, M., Lust, B. and Chiarcos, C.: *Development of Linguistic Linked Open Data Resources for Collaborative Data-Intensive Research in the Language Sciences.* MIT Press, Cambridge, Massachusetts.

Lukasiewicz, T., Straccia, U. (2008). Managing uncertainty and vagueness in description logics for the semantic web. In *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6(4), pp. 291-308.

Maks, I., van der Vliet, H., Görög, A. and Vossen, P. (2013). *User Documentation of Cornetto LMF Lexical Resource for Dutch*. CLARIN Deliverable D9.

McCrae, J., Aguado-de Cea, G., Buitelaar, P., Cimiano, P., Declerck, T., Gomez-Perez, A., Garcia, J., Hollink, L., Montiel-Ponsoda, E., Spohr, D., and Wunner, T. (2012). Interchanging Lexical Resources on the Semantic Web. In *Language Resources and Evaluation*, 46(4), pp. 701–719.

McCrae, J., Labropoulou, P., Gracia, J., Villegas, M., Rodriguez-Doncel, V. and Cimiano, P. (2015). One ontology to bind them all: The META-SHARE OWL ontology for the interoperability of linguistic datasets on the Web. In *Proceedings of 12th Extended Semantic Web Conference (ESWC 2015) Satellite Events*. Portorož, Slovenia, pp. 271–282

Menke, P., Ell, B., Cimiano, P. (2017). On the origin of annotations: A module-based approach to representing annotations in the Natural Language Processing Interchange Format (NIF). In *Applied Ontology*, 12(2), 131-155.

Palmer, M. (2009). Semlink: Linking PropBank, VerbNet and FrameNet. In *Proceedings of the Generative Lexicon Conference (GenLex 2009)*. Pisa, Italy, pp. 9–15.

Rospocher, M., Corcoglioniti, F., and Aprosio, A. P. (2019). PreMOn: LODifing linguistic predicate models. In *Language Resources and Evaluation*, vol. *53*(3), pp. 499–524.

Schuurman, I., Windhouwer, M., Ohren, O. and Zeman, D. (2016). CLARIN concept registry: the new semantic registry. In *Selected Papers from the CLARIN Annual Conference 2015*. Wroclaw, Poland, pp. 62–70.

Swanepoel, P. H. (2015). The design of morphological/linguistic data in L1 and L2 monolingual, explanatory dictionaries: a functional and/or linguistic approach? In *Lexikos*, vol. 25, pp. 353–386.

Verhagen, M., Suderman, K., Wang, D., Ide, N., Shi, C., Wright, J., and Pustejovsky, J. (2016). The LAPPS interchange format. In *Revised Selected Papers of the Second International Workshop on Worldwide Language Service Infrastructure WLSI (2015)*, Kyoto, Japan, pp. 33–47.

Zeman, D. (2008). Reusable Tagset Conversion Using Tagset Drivers. In *Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008*. Marrakech, Morocco.