PreMOn: LODifying Linguistic Predicate Models

Marco Rospocher, Francesco Corcoglioniti, Alessio Palmero Aprosio

Received: date / Accepted: date

Abstract PreMOn is a freely available linguistic resource for exposing predicate models (PropBank, NomBank, VerbNet, and FrameNet) and mappings between them (e.g., SemLink and the Predicate Matrix) as Linguistic Linked Open Data (LOD). It consists of two components: (i) the PreMOn Ontology, that builds on the OntoLex-Lemon model by the W3C Ontology-Lexica Community Group to enable an homogeneous representation of data from various predicate models and their linking to ontological resources; and, (ii) the PreMOn Dataset, a LOD dataset integrating various versions of the aforementioned predicate models and mappings, linked to other LOD ontologies and resources (e.g., FrameBase, ESO, WordNet RDF). PreMOn is accessible online in different ways (e.g., SPARQL endpoint), and extensively documented.

1 Introduction

In the last few years, the use of Semantic Web (SW) technologies and the adoption of the Linked Open Data (LOD) paradigm have gained growing popularity among the Computational Linguistics (CL) community (cf. Chiarcos et al 2013). In particular, initiatives such as the Linguistic Linked Open Data movement have pushed the publishing of CL and Natural Language Processing (NLP) data using the LOD principles.

In this context, in 2015 we started the development of PreMOn (predicate model for ontologies) (Corcoglioniti et al 2016b), a resource exposing predicate models according to SW and LOD best practices. Predicate models, such as PropBank (PB - Palmer et al 2005), NomBank (NB - Meyers et al 2004), VerbNet (VN - Schuler 2005), and FrameNet (FN - Baker et al 1998), provide rich descriptions of predicates abstracting from many linguistic phenomena related to their realization.

1 http://linguistic-lod.org/
2 http://premon.fbk.eu/
in text. Consider the sentence: ‘Steve Jobs sells Pixar to Disney’. Over the years, computation linguistic researchers have invested substantial effort in developing models that associate lexical entries acting as predicates (e.g., the ‘sell’ verb in the example) to semantic classes — i.e., rolesets in NB and PB, verb classes in VN, and frames in FN (e.g., “Commerce Sell”) —, each one conveying a possible predicate sense characterized by a set of semantic roles (e.g., FN ‘Seller’, ‘Buyer’, ‘Goods’) that can be played by predicate arguments in the text (respectively ‘Steve Jobs’, ‘Disney’, ‘Pixar’). Together with Semantic Role Labeling (SRL) tools for the detection of predicates and their arguments in text, and the mappings of resources between different predicate models, such as SemLink (Palmer, 2009) and the Predicate Matrix (Lacalle et al. 2014), predicate models are central to a number of tasks such as Information Extraction, Question Answering, and Natural Language Generation. Among all, predicate models are increasingly used within the SW community for Knowledge Extraction in tools such as NewsReader (Rospocher et al, 2016) and PIKES (Corcoglioniti et al, 2016a), or as the starting point for deriving general-domain ontologies grounded in natural language, such as FrameBase (Rouces et al, 2015), and the ESO (Event Situation Ontology - Segers et al 2015).

PreMOn consists of two components: (i) the PreMOn Ontology, an OWL 2 ontology that models the core concepts of semantic class, semantic role, mapping, and annotation common to all predicate models, as well as their specializations for each model; and, (ii) the PreMOn Dataset, a freely-available 5-star\(^3\) LOD dataset consisting of various releases of PB, NB, VN, FN, and the associated mapping resources, instantiating the PreMOn Ontology conceptual schema. Besides making available with proper URIs (hence, referenceable) all these heterogeneous predicate models — each one having its own terminology, structure, and proprietary release format — via a homogeneous conceptual schema, PreMOn brings several benefits to users of predicate models:

i) it enables intra- and inter-resource querying and analyses, favoring, for instance, debugging of errors (e.g., improper semantic roles used in a given semantic class, or mappings between “wrong” classes) and extension of the resources (e.g., scouting for missing mappings);

ii) it fills the gap for the exploitation of predicate model data in SW-ready applications and annotation frameworks (e.g., NIF - Hellmann et al, 2013), as these need proper identifiers (i.e., URIs) for using such data;

iii) it sets the bases for the extension and proper alignment of predicate models to ontological resources, providing a concrete representation for these mappings that abstracts from their provenance.

We are not aware of other initiatives providing a unified, comprehensive, and queryable view of predicate model data and mappings as the one provided by PreMOn.

Since its debut in late 2015, we provided five major PreMOn releases, each extending the previous one with new predicate model data and mapping resources. Access to the resource is granted in several modalities (SPARQL endpoint, URI dereferencing, ontology and dataset downloads, and a newly released web PreMOn Navigator tool), complemented by an extensive documentation accessible

\(^3\) http://5stardata.info/en/
through the website (e.g., ontology diagrams) and the open-source tool we developed to populate PreMOn from the predicate model source files. Notwithstanding its specificity, we have evidences of substantial usage of the resource, as confirmed by approx. 2.4K dataset downloads and 2.5M queries fired to the PreMOn endpoint over a year.

PreMOn was initially presented in [Corcoglioniti et al. (2016b)]. In this paper, we substantially extend the work presented in [Corcoglioniti et al. (2016b)] along several dimensions by:

- better and more extensively comparing PreMOn with related SW initiatives;
- adding new predicate model versions, namely PB v3.1 (in which unified frames were introduced), VN v3.3, and FN v1.7, thus confirming the capability of PreMOn to cope with predicate model data;
- adding new predicate model mapping resources, namely the Predicate Matrix v1.3, whose n-ary mappings (i.e., mappings involving semantic classes and roles from more that two distinct predicate models) allowed validating PreMOn modeling of predicate model mappings;
- revising the PreMOn Ontology adding the capability to model ontological mappings, i.e., alignment of predicate model data with ontological resources. We validated this capability enriching PreMOn with available mappings to FrameBase v2.0 and ESO v2.0;
- introducing the PreMOn Navigator a web-based interface to navigate and lookup PreMOn data with a browser;
- describing concrete PreMOn application scenarios.

As a result, on a purely numerical level, PreMOn triples sensibly increased from 19.7M (PreMOn v2016a - the version described in [Corcoglioniti et al. 2016b]) to 32.6M (PreMOn v2018a - the version described in this paper).

This paper is structured as follows. Section 2 provides an overview of the resources considered in PreMOn. Section 3 gives an overview of PreMOn, illustrating the PreMOn Ontology with its core concepts. Section 4 describes its population and the creation of the PreMOn Dataset. Section 5 illustrates all the different modalities for accessing PreMOn data, while Section 6 presents some concrete application usage scenarios. Section 7 relates PreMOn with other initiatives in the CL and SW communities. Finally, Section 8 concludes with some final remarks.

2 Background

We briefly overview the resources relevant for PreMOn development: the linguistic resources PreMOn exposes in OWL/RDF (predicate models and their mappings), the ontological model on which PreMOn builds on (OntoLex-Lemon), and the ontological resources to which predicate models are mapped to in PreMOn (FrameBase and ESO). A comparison with other approaches exposing predicate model (and mappings) data in OWL/RDF is reported in Section 7.

PropBank (PB) by [Palmer et al. 2005] is a predicate model for verbs, later extended to other parts-of-speech in [OntoNotes]. PB associates

[^1]: http://verbs.colorado.edu/~mpalmer/projects/ace.html
Table 1: Comparison of predicate models.

<table>
<thead>
<tr>
<th>Aspect</th>
<th>PropBank</th>
<th>NomBank</th>
<th>VerbNet</th>
<th>FrameNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Considered parts-of-speech</td>
<td>verb, noun, adjective</td>
<td>noun</td>
<td>verb</td>
<td>any (9 total)</td>
</tr>
<tr>
<td>Term for semantic classes</td>
<td>roleset</td>
<td>roleset</td>
<td>(sub-)class</td>
<td>frame</td>
</tr>
<tr>
<td>Scope of semantic roles</td>
<td>one or more local to semantic class</td>
<td>exactly one local to semantic class</td>
<td>zero or more global</td>
<td>zero or more local to semantic class (core roles)</td>
</tr>
<tr>
<td>Types of semantic roles</td>
<td>numbered, modifier, secondary agent</td>
<td>numbered, modifier</td>
<td>thematic role (hierarchy)</td>
<td>core + other 3 types</td>
</tr>
<tr>
<td>Semantic class relations</td>
<td>–</td>
<td>–</td>
<td>subclass</td>
<td>inheritance + 8 relations</td>
</tr>
<tr>
<td>Semantic role relations</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>inheritance + 9 relations</td>
</tr>
<tr>
<td>Additional features</td>
<td>mappings to VN</td>
<td>Hmappings to PB, VN</td>
<td>selectional restrictions, syntactic frames</td>
<td>semantic types on classes, roles, lexical units</td>
</tr>
</tbody>
</table>

A lexical entry to one or more semantic classes called *rolesets*. Semantic roles are defined locally to each semantic class and categorized as *numbered arguments* (e.g., Arg0 and Arg1, usually the proto-agent and proto-patient), *modifiers* (ArgM plus a function tag, such as LOC for location), and *secondary agent* (ArgA in OntoNotes). Annotated examples are provided for each semantic class. A summary of PB features, compared to other predicate models, is reported in Table 1.

**NomBank** (NB) by [Meyers et al. (2004)](http://nlp.cs.nyu.edu/meyers/NomBank.html) is a model for noun predicates that closely mirrors PB, in that it associates nouns to (noun-specific) semantic classes also called *rolesets*, and defines semantic roles locally to semantic classes, categorizing them as *numbered arguments* and *modifiers*. Examples are provided for each semantic class.

**VerbNet** (VN) by [Schuler (2005)](http://verbs.colorado.edu/~mpalmer/projects/verbnet.html) is a predicate model for verbs inspired by Levin classes [Levin (1993)]. Semantic classes in VN are organized in a hierarchy of *verb classes*. A verb class is associated to multiple verb lexical entries and to multiple globally defined *thematic roles* (e.g., agent, patient), which describe the semantic roles for that class (with possible selectional restrictions) and form themselves a role hierarchy. Verb classes are also associated to *syntactic frames*. They define how semantic roles can be realized syntactically (e.g., “Agent V Theme”) and are associated to a specification of the conveyed semantics based on logical-like predicates applied to thematic role and event variables (e.g., “cause(Agent, E)”)

Both thematic roles and syntactic frames are inherited by subclasses. An example sentence is provided for each frame of a verb class.

**FrameNet** (FN) by [Baker et al. (1998)](http://framenet.icsi.berkeley.edu/fndrupal/) builds on the theory of frame semantics and provides semantic classes called *frames* that define prototypical situations evoked...
by lexical entries of different parts-of-speech. The pair (frame, lexical entry) is called **lexical unit**. The semantic roles of a frame are called **frame elements** (FEs): core and core unexpressed FEs classify mandatory arguments and characterize a frame, differently from peripheral FEs, while extra-thematic FEs (e.g., “iteration”) situate the frame in a larger context. A set of typically co-expressed FEs form a CoreSet. Several Frames and FE relations are defined, starting from frame inheritance, along which Core FEs (but not unexpressed ones) are propagated. A small subset of frames and FEs is annotated with **semantic types**, that for FEs are used to express selectional constraints. Annotated examples are provided for each lexical unit.

**Predicate models mappings** PB and NB come with some mappings of their semantic classes and roles to corresponding elements in other predicate models (namely, PB to VN and FN, and NB to VN and PB). Additional mappings are provided by two specialized mapping resources: SemLink and PredicateMatrix. *SemLink* by [Palmer (2009)](http://verbs.colorado.edu/semlink/) provides a mapping between the (lexical entry, semantic class) pairs of (i) VN and PB and (ii) VN and FN, as well as between the semantic roles of these resources. *PredicateMatrix* by [Lacalle et al. (2014)](http://adimen.si.ehu.es/web/PredicateMatrix/) is a mapping resource relating semantic classes and semantic roles of VN, PB, and FN. The PredicateMatrix includes the mappings of SemLink and augments them based on the use of automatic techniques leveraging WSD (Word Sense Disambiguation). Mappings to WordNet [Fellbaum (1998)](http://adimen.si.ehu.es/web/MCR) senses, to concepts of the Multilingual Central Repository (MCR) and to additional resources are also provided.

**OntoLex-Lemon** As noted in Section 1, the use of SW technologies and the adoption of the LOD paradigm have already been recognized as particularly beneficial to linguistic resources, leading to the creation of the Linguistic Linked Open Data movement. In that context, the *lemon* (lexical model for ontologies) model by [McCrae et al. (2012)](http://verbs.colorado.edu/semlink/) and its latest development — *OntoLex-Lemon* (McCrae et al., 2017) — by the W3C Ontology-Lexica Community Group, set the basis for modeling the lexical entries referenced in predicate models (e.g., the verb ‘sell’), together with their links (lexical senses) to corresponding ontological concepts.

**Frame-based ontologies** Due to their wide coverage, predicate models and in particular FN have been leveraged for the definition of SW frame-based ontologies, where knowledge is modeled around **semantic frames**, i.e., events, situations, n-ary relations, and other structured entities reified as ontological instances (e.g., a sell event) and connected to related instances via properties (e.g., seller, buyer) specifying their semantic roles in the frame (so-called Neo-Davidsonian representation — [Parsons, 1990]). Two such ontologies are ESO (Event Situation Ontology — Ségers et al. 2015) and FrameBase (Rouces et al. 2015), whose frame classes and properties are obtained by revising/extending FN frames and roles, possibly combined with other linguistic resources such as WordNet, which in FrameBase is used to split FN frames in synset-based **micro-frame** classes.
3 The PreMON Ontology

The ontological schema of PreMON is defined in the PreMON Ontology. Organized in a core module and several predicate model-specific specializations, the PreMON Ontology builds on OntoLex-Lemon (McCrae et al., 2017), a model enabling rich linguistic grounding for ontologies. As summarized in Fig. 1 (white-filled concepts) OntoLex-Lemon sets the basis for modeling the lexical entries (class `ontolex:LexicalEntry`) referenced in predicate models (e.g., the verb ‘sell’), together with their associated lexical forms, and allows relating entries to the ontological entities they denote (classes, properties, individuals) using the `ontolex:LexicalSense` reified relation.

![Fig. 1 The PreMON Ontology and its grounding in OntoLex-Lemon.](image)

3.1 Semantic Classes and Roles

At the core of the PreMON Ontology are classes `pmo:SemanticClass` and `pmo:SemanticRole` (green-filled in Fig. 1). `pmo:SemanticClass` homogeneously represents the semantic classes from the various predicate models, abstracting over model specific peculiarities. That is, individuals of this class correspond to rolesets in PB and NB (e.g., `pm:nb10-seller.01` and `pm:pb215-sell.01` in Fig. 2), verb classes in VN (e.g., `pm:vn32-give-13.1-1`), and frames in FN (e.g., `pm:fn16-commerce-sell`). An instance of `pmo:SemanticClass` typically has (via property `pmo:semRole`) a number of `pmo:SemanticRoles`, representing, from a semantic point of view, the roles the arguments of that `pmo:SemanticClass` can play. For instance, the triple (cfr. Fig. 2)

\[
\text{pm:fn16-commerce-sell} \text{ pmo:semRole} \text{ pm:fn16-commerce-sell@seller}.
\]

states that `pm:fn16-commerce-sell` has the semantic role `pm:fn16-commerce-sell@seller`. Note that `pmo:SemanticClass` is defined as subclass of `ontolex:LexicalConcept`, as we see `pmo:SemanticClasses` as essentially informal concepts rather than well

---

11 We provide here a concise description of PreMON. Further details are available on the PreMON website, the ontology LODE documentation or Corcoglioniti et al. (2016b).

12 pm: http://premon.fbk.eu/resource/ is the namespace of PreMON Dataset; the fragments nb10, pb215, and so on identify the resource and its version (e.g., NB 1.0, PB 2.1.5).
defined concepts of a formal ontology (although some ontologies are derived from them, e.g., FrameBase and ESO), similarly to WordNet synsets that can also be seen as *ontolex:LexicalConcepts*. This modeling decision prevents the direct reuse of *OntoLex-Lemon* concepts for mapping to external ontologies, and sets *PreMOn* apart from *lemonUby* [Eckle-Kohler et al. 2015] that links semantic classes to *ontolex:LexicalSenses*, as discussed in Section 2. Being *ontolex:LexicalConcepts*, *pmo:SemanticClasses* inherit the link to lexical entries as well as the link (via *ontolex:isConceptOf*) to the ontological entities formalizing them, typically event classes. Properties *pmo:classRel*, *pmo:roleRel*, and their resource-specific subproperties, are introduced to express the relations between elements at each level, such as subtyping, and predicate and role inheritance. Additional resource-specific classes (e.g., *pmovn:ThematicRole*, in light blue in Figure 2) and properties (e.g., *pmovn:thematicRole*) further characterize important aspects of each predicate model, like semantic role commonalities.

### 3.2 Mappings between Predicate Models

Mappings between different predicate models are practically relevant but cannot be expressed using only the classes above, as they are often defined (e.g., in *SemLink* and *PredicateMatrix*) in terms of (*pmo:SemanticClass*, *ontolex:LexicalEntry*) pairs. To model these pairs, one could reuse the notion of *ontolex:LexicalSense* (see Fig. 1). However, its formalization as reified relation in *OntoLex-Lemon* would require the existence of *exactly* one ontological entity for each (*ontolex:LexicalConcept*, *ontolex:LexicalEntry*) pair, a strong constraint that we do not necessarily need for
our purposes. Therefore, we introduce the loosely axiomatized pmo:Conceptualization class (filled in light red in Figure 1) together with other mapping related classes. 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.

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. Figure 2 shows an example of mapping (pm:mapping1356) between two pmo:Conceptualizations, one from NB (pm:co-n-seller-nb10-seller.01) and one from PB (pm:co-v-sell-pb215-sell.01). We rely on this set-like modeling, since mappings are not necessarily represented as binary relations in predicate mapping resources: e.g., in the PredicateMatrix, each row represents the mapping of a semantic role / lexical entry pair over the different resources (e.g., (13.1-1-agent, deal) in VN, (sell.01-arg0, sell) in PB, (Commerce_Sell-seller, sell) in FN) as well as the corresponding WordNet verb sense. Reifying the n-ary mapping relation also allows us, if needed, to further characterize each single mapping, asserting additional information such as confidence and reliability. Moreover, it is possible to further specialize mappings (e.g., to model mappings holding only in one direction, from a resource to another one, or to represent different types of relationships among the members of the mapping) by subtyping the pmo:Mapping class or the property (pmo:item) relating a pmo:Mapping to its members.

3.3 Mappings to Ontological Resources

One of the motivations behind the development of PreMO was to explicitly represent, in a SW-compliant way, mappings of predicate model semantic classes (roles) to ontology classes (properties), so to indirectly provide a formal semantics for predicate model elements. OntoLex-Lemon provides a mechanism for mapping lexical entries to ontological entities via the ontolex:LexicalSense concept. PreMO extends this mechanism with property pmo:ontologyMatch (cf. Fig. 1), which allows to map pmo:SemanticClasses, pmo:SemanticRoles, and pmo:Conceptualizations to resources (e.g., classes, properties) in ontologies. Figure 2 shows some instantiations of these mappings to FrameBase and ESO. Note that similarly to the mappings between predicate models, in some cases also ontological mappings can be defined on (pmo:SemanticClass, ontolex:LexicalEntry) pairs (and hence, pmo:Conceptualizations) as shown by the mapping of pm:co-v-sell-fn16-commerce_sell to class fbframe:Commerce_sell.sell.verb in FrameBase.

13 In case of a (pmo:SemanticClass, ontolex:LexicalEntry) pair mapped to multiple ontology entities (e.g., in FrameBase and ESO), using ontolex:LexicalSense instead of pmo:Conceptualization would require all those entities to be owl:sameAs one another, a strong inference generally not anticipated nor intended by the authors of mappings (especially if defined independently).

14 https://www.w3.org/community/ontolex/wiki/Final_Model_Specification#Lexical_Sense_26_Reference
3.4 Annotations

Predicate models are typically complemented by examples showing concrete occurrences of semantic classes and roles in text. More generally, a text can be annotated with semantic classes and roles as a result of manual or automatic SRL.

The PreMOn Ontology provides some common primitives (filled in light yellow in Figure 1), based on the NLP Interchange Format (NIF) by Hellmann et al. (2013), which aim at properly modeling the heterogeneous annotations of a text for different predicate models. NIF introduces nif:Context is a particular subclass of nif:String, representing a whole string of text. Any substring (itself a nif:String) has a nif:referenceContext relation to the nif:Context individual representing the whole text containing it.

To specifically model the aforementioned examples complementing predicate models, we introduce pmo:Example, subclass of nif:Context, to represent the string associated with the example. The occurrence of a ontolex:LexicalEntry, pmo:SemanticClass, or pmo:SemanticRole in a nif:Context is denoted by an instance of nif:Annotation, related to the given ontolex:LexicalEntry, pmo:SemanticClass, or pmo:SemanticRole via property pmo:valueObj (the value attached to the annotation), and to the nif:Context instance via property nif:annotation. If detailed information on the specific span of text (i.e., substring) denoting the ontolex:LexicalEntry, pmo:SemanticClass, or pmo:SemanticRole is available (e.g., FN provides the specific offsets of lexical units, frames, and frame elements, in the example text) an additional instance of pmo:Markable, subclass of nif:String, is created and linked to the specific nif:Annotation and nif:Context via properties nif:annotation and nif:referenceContext, respectively. As the same nif:Context may contain multiple nif:Annotations referring to one or more semantic classes and their corresponding roles, an additional pmo:AnnotationSet instance is created to cluster annotations from the same predicate structure.

3.5 Specializing the PreMOn Ontology

While the PreMOn Ontology Core Module provides an homogeneous abstraction over heterogeneous predicate models, additional ontology modules, specializing or extending the PreMOn Ontology core elements, can be defined to properly capture and model resource-specific aspects (including terminology) in a way compatible with the underlying PreMOn Ontology assumptions. We developed four ontology modules, one for each predicate model: PB, NB, VN, and FN. A detailed description of each module (including ontology diagrams highlighting the main classes and properties, and the relation with the Core Module) is provided in Appendix; c.f., PB (A1), NB (A2), VN (A3), and FN (A4). Additional information is provided on the PreMOn website.

4 Populating the PreMOn Ontology: the PreMOn Dataset

The PreMOn Dataset instantiates the PreMOn Ontology with the data of various (versions of) predicate models and mapping resources available. The instantiation is automatically performed via PreMOnitor, an open-source Java command-line
tool available on GitHub\footnote{https://github.com/dkmfbk/premon} (source code) and the PreMOn website (pre-compiled binaries). The tool applies pluggable, resource-specific converters to the original distribution files of each resource, instantiating the proper individuals and assertions according to the PreMOn Ontology. This consists in: (i) parsing the given files, which typically are in XML or CSV format; and, (ii) iterating over the role-sets/verb classes/frames in the original predicate model files, instantiating the appropriate PreMOn instances and triples. In doing this, related content such as the corresponding arguments/roles/frame elements, examples, and mappings, potentially distributed over several files, is also instantiated. While sharing similar logics, tailored conversion strategies were implemented in PreMOnitor converters for each predicate model, to account for the model specificities. E.g., in VN, semantic roles (with selectional constraints) and frames have to be propagated from a class to its subclasses, unless redefined in the latter. Analogously, the instantiation of \texttt{pmo:SemanticRole} for PB/NB modifier arguments requires creating an individual for each \texttt{⟨roleset,argument⟩} pair, as no information is provided on which arguments a predicate may have (besides explicit occurrence in frame files, in which case a module specific semantic role attribute, namely \texttt{pmopb:core} (PB) and \texttt{pmonb:core} (NB), is instantiated —cf. Appendix A1-A2).

If available in the distribution files, mappings to non-predicate model resources (e.g., WordNet synsets, OntoNotes groupings) are also extracted. OWL 2 RL inference, VOID\footnote{https://www.w3.org/TR/void/} structural statistics extraction and some cross-resource cleanup (e.g., for dropping inconsistent mapping, or references to wrong/non-existing resources) are applied to extracted triples, leveraging RDFpro (Corcoglioniti et al, 2015) for RDF processing.

Due to the implemented conversion strategy, the whole content of the original resources is instantiated in the PreMOn Dataset: minor differences can be observed due to the removal of redundant or wrong information during the aforementioned cleanup process. Moreover, several checks are performed (released as scripts within the PreMOnitor package) to ensure the correct instantiation of the PreMOn Dataset both with respects to the PreMOn Ontology schema (e.g., to check that functionality constraints on properties are met on the instantiated data) and to the original resources’ source files (e.g., to check that the expected \texttt{pmo:SemanticClass} and \texttt{pmo:SemanticRole} instances are created).

To track provenance at a coarse-grained level and allow querying only data of specific predicate models (e.g., using SPARQL clauses \texttt{FROM} and \texttt{FROM NAMED}), triples are placed in distinct named graphs identified by the resource name and version (e.g., \texttt{pm:fn16} for FN v1.6). Examples and related triples are placed in further separated named graphs (e.g., \texttt{pm:fn16-ex}), and their extraction can be toggled at runtime.

As new releases of a predicate model or a mapping resource typically share the same structure of the source files of previous versions, populating them in PreMOn generally boils down to running the specific PreMOnitor converter on the source files for the new versions, without any changes on the code. This makes the update and maintainability of PreMOn a reasonably straightforward process. Moreover, the pluggable nature of PreMOnitor facilitates the integration of converters for predicate model newcomers.
Table 2 PreMOn Dataset Releases

<table>
<thead>
<tr>
<th>Release (date)</th>
<th>Predicate Models</th>
<th>Mapping Resources</th>
<th>Mappings to Ontological Resources</th>
<th>Triples</th>
</tr>
</thead>
<tbody>
<tr>
<td>v2015a (2015 Nov 12)</td>
<td>pb17; pb215; nb10</td>
<td>-</td>
<td>-</td>
<td>13646439</td>
</tr>
<tr>
<td>v2016a (2016 Mar 23)</td>
<td>pb17; pb215; nb10; fn15;</td>
<td>sl122c</td>
<td>-</td>
<td>19770136</td>
</tr>
<tr>
<td>v2016b (2016 May 4)</td>
<td>pb17; pb215; nb10; fn15; fn16</td>
<td>sl122c; pm13</td>
<td>-</td>
<td>20170906</td>
</tr>
<tr>
<td>v2017a (2017 May 1)</td>
<td>pb17; pb215; nb10; vn32; fn15; fn16</td>
<td>-</td>
<td>-</td>
<td>31328485</td>
</tr>
<tr>
<td>v2018a (2018 Feb 18)</td>
<td>pb17; pb215; pb31; nb10; vn32; fn15; fn15; fn16; fn17</td>
<td>s122c; pm13</td>
<td>-</td>
<td>32611819</td>
</tr>
</tbody>
</table>

Over the last couple of years, we applied PreMOnitor on a number of different resources to produce four releases of the PreMOn Dataset. These resources include:

- predicate models: PB v1.7 (pb17 hereafter), v2.1.5 (pb215), and v3.1 (pb31); NB v1.0 (nb10); VN v3.2 (vn32); VN v3.3 (vn33); FN v1.5 (fn15), v1.6 (fn16), and v1.7 (fn17);
- predicate mapping resources: SemLink 1.2.2c (sl122c) and Predicate Matrix v1.3 (pm13), providing mapping among semantic classes and roles, including mappings from VN classes to WordNet 3.1 (wn31) synsets and OntoNotes 5 (on5) groupings.
- mappings to ontological concepts (for semantic classes, roles, and conceptualizations): FrameBase v2.0 (fb20) and ESO v2.0 (eso20), with mapping extracted from these resources and from the PB/NB to FrameBase mappings released with PIKES (Corcoglioniti et al, 2016a).

Table 2 summarizes the resources included in each PreMOn Dataset release and the total number of triples it contains, highlighting how it has substantially grown since its first releases. We remark that the mappings to WordNet, or to ontologies such as FrameBase and ESO, enable to link PreMOn to other (Linguistic) LOD resources, thus making PreMOn a proper citizen of the Linguistic LOD cloud.

Focusing on the latest PreMOn Dataset release (v2018a), Table 3 reports the number of individuals (e.g., of pmo:SemanticClasses) and triples for the considered predicate models and their versions. Table 4 reports the numbers of mappings, which may be lower than the ones in the original resources as we drop mappings involving non-existing semantic classes, roles, or conceptualizations (e.g., because removed/reorganized in the newer versions of the resources, released after the mapping was defined).

5 Accessing PreMOn

Access to PreMOn content is provided in various modalities, all publicly available from the PreMOn website, in compliance with the 5-star LOD best practices.

**URI dereferencing** All PreMOn URIs can be dereferenced, obtaining different representations based on HTTP content negotiation as per LOD publishing best prac-

---

17 Version described in [Corcoglioniti et al, 2016b](https://www.w3.org/DesignIssues/LinkedData.html)

18 https://www.w3.org/DesignIssues/LinkedData.html
### Table 3 PreMON Dataset v2018a — General Statistics.

<table>
<thead>
<tr>
<th></th>
<th>nb10</th>
<th>pb17</th>
<th>pb215</th>
<th>vn32</th>
<th>vn33</th>
<th>fn15</th>
<th>fn16</th>
<th>fn17</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sem. classes</td>
<td>5576</td>
<td>6198</td>
<td>9208</td>
<td>10687</td>
<td>484</td>
<td>599</td>
<td>1018</td>
<td>1204</td>
<td>1220</td>
</tr>
<tr>
<td>Sem. roles</td>
<td>76241</td>
<td>164581</td>
<td>245135</td>
<td>283500</td>
<td>1396</td>
<td>1769</td>
<td>9633</td>
<td>11251</td>
<td>11428</td>
</tr>
<tr>
<td>Conceptualize.</td>
<td>5576</td>
<td>6199</td>
<td>9208</td>
<td>15567</td>
<td>6338</td>
<td>6814</td>
<td>11887</td>
<td>13247</td>
<td>13561</td>
</tr>
<tr>
<td>Lexical entries</td>
<td>4704</td>
<td>5199</td>
<td>7671</td>
<td>13030</td>
<td>4402</td>
<td>4569</td>
<td>9413</td>
<td>10107</td>
<td>10345</td>
</tr>
<tr>
<td>Examples</td>
<td>9583</td>
<td>10221</td>
<td>15835</td>
<td>22598</td>
<td>1396</td>
<td>1589</td>
<td>153589</td>
<td>174425</td>
<td>177417</td>
</tr>
<tr>
<td>Annotation sets</td>
<td>9583</td>
<td>10221</td>
<td>15835</td>
<td>22598</td>
<td>1396</td>
<td>1589</td>
<td>154573</td>
<td>196610</td>
<td>201729</td>
</tr>
<tr>
<td>Core triples</td>
<td>518200</td>
<td>688715</td>
<td>1050825</td>
<td>1307827</td>
<td>247885</td>
<td>312277</td>
<td>332131</td>
<td>413773</td>
<td>429424</td>
</tr>
<tr>
<td>Example triples</td>
<td>339499</td>
<td>429947</td>
<td>653395</td>
<td>881774</td>
<td>23800</td>
<td>27204</td>
<td>575213</td>
<td>729630</td>
<td>729725</td>
</tr>
<tr>
<td>Inferred triples</td>
<td>252843</td>
<td>410033</td>
<td>618886</td>
<td>743187</td>
<td>23800</td>
<td>27204</td>
<td>575213</td>
<td>729630</td>
<td>729725</td>
</tr>
<tr>
<td>Total triples</td>
<td>1110542</td>
<td>1528695</td>
<td>2323106</td>
<td>2932788</td>
<td>1110542</td>
<td>423935</td>
<td>6634669</td>
<td>8172610</td>
<td>8374932</td>
</tr>
</tbody>
</table>

### Table 4 PreMON Dataset v2018a — Mapping Statistics.

<table>
<thead>
<tr>
<th>mapped resources</th>
<th>source</th>
<th># valid mappings</th>
<th># discarded mappings</th>
<th>concept.</th>
<th>class</th>
<th>role</th>
<th>concept.</th>
<th>class</th>
<th>role</th>
</tr>
</thead>
<tbody>
<tr>
<td>nb10-to-vn32</td>
<td>nb10</td>
<td>1081</td>
<td>1833</td>
<td>3410</td>
<td>773</td>
<td>21</td>
<td>950</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nb10-to-pb17</td>
<td>nb10</td>
<td>3474</td>
<td>3474</td>
<td>9841</td>
<td>12</td>
<td>12</td>
<td>397</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nb10-to-pb215</td>
<td>nb10</td>
<td>3473</td>
<td>3473</td>
<td>9840</td>
<td>13</td>
<td>13</td>
<td>398</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pb17-to-vn32</td>
<td>pb17</td>
<td>2930</td>
<td>4085</td>
<td>7967</td>
<td>1454</td>
<td>17</td>
<td>1251</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pb215-to-vn32</td>
<td>pb215</td>
<td>4862</td>
<td>6909</td>
<td>13573</td>
<td>2310</td>
<td>32</td>
<td>1921</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fn15-to-pb17</td>
<td>fn15</td>
<td>1676</td>
<td>1896</td>
<td>-</td>
<td>296</td>
<td>76</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pb31-to-vn32</td>
<td>pb31</td>
<td>5800</td>
<td>5675</td>
<td>13369</td>
<td>3607</td>
<td>40</td>
<td>108</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fn15-to-pb31</td>
<td>fn15</td>
<td>2440</td>
<td>2592</td>
<td>618</td>
<td>126</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vn32-to-vn31</td>
<td>vn32</td>
<td>5248</td>
<td>-</td>
<td>5248</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>on5-to-vn32</td>
<td>on5</td>
<td>3357</td>
<td>-</td>
<td>3357</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vn33-to-vn31</td>
<td>vn33</td>
<td>5513</td>
<td>-</td>
<td>5513</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fn15-to-fn16</td>
<td>fn16</td>
<td>1014</td>
<td>9572</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fn16-to-fn17</td>
<td>fn17</td>
<td>-</td>
<td>1202</td>
<td>11211</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vn32-to-wn31</td>
<td>pn13</td>
<td>8354</td>
<td>-</td>
<td>8354</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pb17-to-vn32</td>
<td>pm13</td>
<td>4947</td>
<td>4946</td>
<td>11307</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pb17-to-vn31</td>
<td>pm13</td>
<td>7944</td>
<td>-</td>
<td>7944</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fn15-to-vn32</td>
<td>pm13</td>
<td>4273</td>
<td>2963</td>
<td>6145</td>
<td>204</td>
<td>74</td>
<td>38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fn15-to-pb17</td>
<td>pm13</td>
<td>3943</td>
<td>6189</td>
<td>13086</td>
<td>219</td>
<td>131</td>
<td>1109</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fn15-to-vn31</td>
<td>pm13</td>
<td>6562</td>
<td>-</td>
<td>6562</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pb215-to-vn32</td>
<td>sl122c</td>
<td>5416</td>
<td>5561</td>
<td>12422</td>
<td>175</td>
<td>30</td>
<td>129</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pb17-to-vn32</td>
<td>sl122c</td>
<td>4518</td>
<td>4631</td>
<td>10088</td>
<td>1073</td>
<td>960</td>
<td>2463</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fn15-to-vn32</td>
<td>sl122c</td>
<td>3617</td>
<td>1715</td>
<td>1904</td>
<td>121</td>
<td>1</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fn15-to-fn20</td>
<td>fn20</td>
<td>11887</td>
<td>-</td>
<td>9633</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>nb10-to-fn20</td>
<td>PIKES</td>
<td>596</td>
<td>596</td>
<td>483</td>
<td>-</td>
<td>-</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pb215-to-fn20</td>
<td>PIKES</td>
<td>2222</td>
<td>2222</td>
<td>2865</td>
<td>-</td>
<td>-</td>
<td>211</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fn16-to-eso20</td>
<td>eso20</td>
<td>-</td>
<td>92</td>
<td>277</td>
<td>-</td>
<td>1</td>
<td>645</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>107776</td>
<td>61068</td>
<td>146993</td>
<td>11160</td>
<td>1534</td>
<td>10078</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SPARQL endpoint PreMON data can be programmatically queried using the dedicated SPARQL endpoint, powered by Virtuoso. A simple interface (see Figure 3) is also provided to manually query the endpoint using the browser.

PreMON downloads The PreMON Ontology is distributed as separate OWL modules (core, PB, NB, VN, FB), with owl:imports taking care of their dependencies; a zip package containing all the required OWL files suitable for import in ontology editors like Protégé is provided too. Also each single component of the PreMON 19

19 http://lodview.it/
Dataset can be downloaded separately, with each predicate model and mapping resources distributed with three packages — core definitions, examples, and VOID statistics — available both with and without OWL2RL inferences (computed with RDFpro using the PreMOn, OntoLex-Lemon and NIF TBox definitions).

PreMOn Navigator In addition to the traditional LOD access modalities, users can browse the PreMOn data with the PreMOn Navigator, a web UI that allows an easy lookup of lexical entries and semantic classes. Entries are organized in toggleable vertical tabs, and by clicking on each of them the corresponding RDF content is retrieved and rendered with LodView. A text box allows to quickly filter-while-typing PreMOn data by matching the textual content provided by the user. Figure 4 shows the navigation of PreMOn data with the PreMOn Navigator.

PreMOn is complemented with an extensive documentation, accessible from the website, explaining how to use the data and software.

6 Application Scenarios

By making available all heterogeneous predicate models via a homogeneous conceptual schema, PreMOn enables several applications, among which:

Knowledge Extraction NLP and Knowledge Extraction tools can exploit PreMOn to express semantic class and role annotations in RDF in a way compatible with the increasingly used NIF. Indeed, the formalization in PreMOn of the predicate model examples shows a first concrete case of predicate model annotation data exposed according to NIF principles, as reported in the following example:
Fig. 4 Browsing PreMO with the PreMO Navigator.

```sparql
pm:fn16-ex_823971 a pmo:Example, nif:Context ;
  nif:anchorOf 'The shoe retails for 79.99'@en ;
  nif:annotation pm:fn16-ex_823971_annSet_9_pred ;
  nif:annotation pm:fn16-ex_823971_annSet_9_arg1.

pm:fn16-ex_823971_annSet_9 a pmo:AnnotationSet ;
  pmo:item pm:fn16-ex_823971_annSet_9_pred ;
  pmo:item pm:fn16-ex_823971_annSet_9_arg1.

pm:fn16-ex_823971_annSet_9_pred a nif:Annotation ;
  pmo:valueObj pm:fn16-commerce_sell , pm:co-v-retail-fn16-commerce_sell .

pm:fn16-ex_823971_annSet_9_arg1 a nif:Annotation ;
  pmo:valueObj pm:fn16-commerce_sell@goods .

pm:fn16-ex_823971?char=9,15 a nif:String, pmo:Markable ;
  nif:annotation pm:fn16-ex_823971_annSet_9_pred ;
  nif:referenceContext pm:fn16-ex_823971 ;
  nif:anchorOf 'retail'@en ;
  nif:beginIndex '9'ˆˆxsd:int ; nif:endIndex '15'ˆˆxsd:int .

pm:fn16-ex_823971?char0,7 a nif:String, pmo:Markable ;
  nif:annotation pm:fn16-ex_823971_annSet_9_arg1 ;
  nif:referenceContext pm:fn16-ex_823971 ;
  nif:anchorOf 'The shoe'@en ;
  nif:beginIndex '0'ˆˆxsd:int ; nif:endIndex '7'ˆˆxsd:int.
```
Intra Predicate Model Analyses  The exposing of predicate model data as OWL / RDF, amenable to the use of reasoning, querying and other SW data processing techniques, opens up new opportunities for analyzing, validating, and possibly cleaning up predicate model data. For instance, the capability to store multiple releases of the same resource (via named graphs) enables to analyze, by means of queries, how a predicate model has changed, in terms of its content (e.g., changes of semantic classes and related lexical entries), from one release to another. This could be particularly helpful for predicate model users, especially when a new version of a predicate model is released, and no explicit, detailed documentation of the changes between the new version and the previous one is provided. Concretely, this is the case we faced when comparing the content of v3.3 to v3.2 of VN. In particular, we wanted to understand how verb classes changed across the two releases, especially in terms of class members. To support our analysis, we performed queries like the one in Figure 5 which leverages the VN class member mappings to WordNet synsets, in order to compare verb classes sharing the same members in different VerbNet versions. For instance, we see that some members of class accept-77 in VN v3.2 are now in the revised accept-77-1 class in VN v3.3 while others are in substantially different classes (e.g., admire-31.2). While the results of the query in Figure 5 cannot be interpreted as precise mappings between verb classes of the considered VN releases, query-enabled analyses like this are an effective tool to support the inspection of changes between different predicate model versions.

PIKES is already using PreMon identifiers for SRL annotations. (cf. http://pikes.fbk.eu/)

We observed the presence of lexical entries with different VN conceptualizations but the same WordNet synset, which implies that the synset alone is insufficient to disambiguate among VN classes/conceptualizations and thus unambiguously relate them across VN versions.
Cross Predicate Model Analyses  Thanks to PreMOn Ontology homogeneous representation of different predicate models, queries over PreMOn data can be fired to simultaneously access all the different predicate models, in order to systematically check their content and the mappings between them.

Consider for instance the following SPARQL query:

```sparql
SELECT DISTINCT ?lexEnt (COUNT(?resource) AS ?n)
WHERE {
  { SELECT DISTINCT ?lexEnt ?resource WHERE { GRAPH ?resource { ?lexEnt ontolex:evokes [ a pmo:SemanticClass ] } } }
  FILTER NOT EXISTS { ?conc pmo:evokingEntry ?lexEnt.
    ?mapping a pmo:Mapping; pmo:item ?conc }
}
GROUP BY ?lexEnt
ORDER BY DESC (?n) ?lexEnt
```

Such query asks to retrieve lexical entries (?lexEnt) evoking semantic classes in different predicate models for which no mapping exists. Among the results of such query, the lexical entry pm:v-pan is defined in 7 different predicate model datasets, but no mapping between semantic classes insisting on it is defined in any of the available resources. The results of such query may suggest new possible mappings between predicate models. Another more complex example is reported in the previous Figure 3, which shows a similar query that returns pairs of conceptualizations (extracted as (?lexEnt,?pbRoleset) and (?lexEnt,?fnFrame) pairs) in PB v2.1.5 and FN v1.5 that insist on the same lexical entry (?lexEnt) and have no existing mappings. Each result is a mapping suggestion and, to help judging the possible mapping, it contains the definitions of the PB roleset and FN lexical unit, and the number of annotated examples and core roles for both conceptualizations.

Similarly, new (implicit) mappings can be suggested by navigating chains of existing mappings, possibly from different mapping sources, via SPARQL property paths. Mappings themselves can also be exploited, together with WordNet RDF data linked in PreMOn, to find synonymous lexical entries that can extend the specific lexicon of one of the resources. Concerning data validation and cleaning, we already implemented a simple technique for identifying and removing inconsistent mappings when instantiating a PreMOn Dataset with PreMOnitor.

Further enhanced predicate model analyses can be performed by exploiting datasets annotated with multiple predicate model, i.e., same text examples with annotations from more than one predicate model, either manually or automatically produced (i.e., processing the textual corpus with SRL tools annotating with different predicate models, such as Mate Tools, Semafor and Boxer). By exposing these annotations in NIF with PreMOn (see “Knowledge Extraction” at the beginning of this section), and injecting them in the same triple store with PreMOn data, queries combining both annotated data and predicate model data can be executed. Such queries can help better assess how the same text gets annotated according to different predicate models, easily filtering on some lexical

---

24 [http://www.cs.cmu.edu/~ark/SEMAFOR/](http://www.cs.cmu.edu/~ark/SEMAFOR/)
entity, semantic class or semantic role, thus favoring predicate model comparisons otherwise difficult to perform.

Mapping of Predicate Models to Ontological Resources PreMOn shares OntoLex-Lemon’s goal to provide proper linguistic grounding for ontologies. A concrete example is the grounding of FrameBase and ESO on PreMOn representation of the linguistic information for predicates, provided in the latest PreMOn releases. Similarly, other event and frame ontologies can be grounded on PreMOn data, thus supporting the development of comprehensive catalogs, from the linguistic to the knowledge level, of event and event participant types.

Besides these possible usages of the resource, the number of downloads (2.4K/year) and SPARQL queries (2.5M/year) fired to the PreMOn endpoint leave open the possibilities of other usages we may not be aware of.

7 Related Works

We compare PreMOn to other approaches exposing (or capable of exposing) predicate model data in OWL/RDF.

Predicate models in RDF/OWL Few attempts have been made for representing predicate models in RDF. To the best of our knowledge, no RDF version of PB and NB exists. An ontological model of VN is presented in [Gangemi] (2010) but no RDF data is available. An RDF version of FN 1.5 is available but its schema is not aligned to lemon and instead closely mirrors the structure and naming used in FN frame files. Finally, some outdated versions of VN and FN (but no PB and NB) have been exposed in RDF/lemon as part of lemonUby [Eckle-Kohler et al. 2015], a lemon version of Uby, but several modeling decisions are no more aligned with the latest developments of OntoLex-Lemon by the W3C Ontology-Lexica Community Group. lemonUby also provides some partial support for linking FN and VN, but only at the level of lemon LexicalSenses (e.g., from the lexical unit of a frame to a member of a class). PreMOn aims at filling this gap, providing a LOD-compliant and OntoLex-Lemon-grounded, faithful and complete view of existing predicate models and their mappings, adopting an homogeneous RDF/OWL representation and making their data referenceable and queryable to the benefit of all the researchers and practitioners using it. Furthermore, PreMOn supports a proper modeling of mappings both at the level of semantic classes/conceptualizations and semantic roles.

OntoLex-Lemon In developing OntoLex-Lemon, the W3C Ontology-Lexica Community Group has made some sharp separation between what is covered in OntoLex-Lemon, and what is outside the model. In particular, the group decided to represent (cf. OntoLex-Lemon synsem module) syntactic frames and arguments, and their

---

26 http://ontologydesignpatterns.org/ont/framenet/html/
27 E.g., frames/verb classes and lexical units/class members both modelled as lemon LexicalSenses and owl:sameAs links between syntactic and semantic arguments (thus implying they are the same).
syntactic dependency, leaving outside the model aspects related to the intensional / non ontologically formalized semantics of predicates (a.k.a. semantic frames) and delegating their formalization to external ontologies (via synsem:OntoMap). This debated decision raised an unsolved issue regarding the relation of OntoLex-Lemon with existing frame-like resources such as FrameNet, which define predicates and roles at the semantics level, but are not ontologies (cf. the ontological analysis of FrameNet by Ovchinnikova et al [2010]). Our work contributes to address this particular issue, providing a concrete proposal to uniformly represent the semantics of predicates and roles as defined by different existing resources. This representation is grounded in OntoLex-Lemon, reusing its vocabulary wherever possible and, in line with its goals, supporting the mapping of predicates and roles to existing frame-inspired ontologies such as FrameBase and ESO.

Frame-based ontological resources While based on predicate models, frame-based ontologies such as FrameBase and ESO aim at (formally) modeling the real-world events, situations and relations expressed by predicates, rather than modeling the predicates themselves as they occur in text. Therefore, aspects of predicate models that are purely linguistic are ignored, and the original structure of semantic classes and roles may be altered (e.g., by fusing or introducing new intermediate classes) to obtain sound ontologies. In contrast, PreMON aims at providing a faithful ontological representation of predicate models as they are, treating semantic classes as informal (intensional) concepts, capturing structural similarities among predicate models and, whenever there exist mappings from semantic classes (roles) to frame-based ontology classes (properties), representing them to indirectly provide a semantics for predicate model elements. Thus, PreMON stands at an intermediate level between OntoLex-Lemon content and ontologies such as FrameBase and ESO. Furthermore, thanks to the PreMON mappings, FrameBase and ESO ontological classes and properties can be (operationally) related to the semantic classes and roles in all the available predicate models.

Framester (Gangemi et al [2016]) is a recent (released after PreMON) OWL/RDF resources aiming at integrating and linking different linguistic resources. Among them, it includes WordNet, BabelNet, FN, and VN. Framester is centered around FN, and all other resources are somehow connected to it. Framester adopts different modeling choices to represent FN in OWL than PreMON: frames are modelled as OWL classes, while frame elements are encoded as object properties. In PreMON, instead, both are modelled as instances of distinct classes, one characterizing semantic classes and the other semantic roles, thus making PreMON more suitable to represent predicate model annotations produced by SRL tools (e.g. Semafor) in OWL/RDF frameworks such as NIF [Hellmann et al 2013]. Furthermore, PreMON provides a common abstraction of predicates and roles that encompasses distinct predicate models: this way, PreMON enables to effectively query and relate data from different resources. Moreover, Framester currently includes and maps only one (outdated) version per resource (e.g., FN 1.5 and VN 3.1), while PreMON enables to gather different, up-to-date versions for each re-

\[28\] Cf. the group’s mailing list thread: https://lists.w3.org/Archives/Public/public-ontolex/2015Jul/0045.html

\[29\] Cf. the group’s mailing list thread: https://lists.w3.org/Archives/Public/public-ontolex/2014Jul/0021.html
source (e.g., FN 1.5, 1.6, and 1.7; VN 3.2 and 3.3), thus enabling also queries comparing different releases.

8 Concluding Remarks

In this paper we presented the latest development of PreMOn, a Linguistic LOD resource firstly released in 2015. PreMOn is a freely available linguistic resource for exposing predicate models (PropBank, NomBank, VerbNet, and FrameNet) and mappings between them (SemLink, Predicate Matrix) and to ontologies (FrameBase, ESO) as Linked Open Data. Besides keeping it periodically updated with new predicate model releases, we plan to further extend and align PreMOn with recent Linguistic LOD initiatives such as Framester (Gangemi et al, 2016), a frame-oriented hub for linguistic data, and AMR-LD (Burns et al, 2016), the LOD version of the Abstract Meaning Representation corpus, which is strongly bound to PB.

Acknowledgements The authors would like to acknowledge the contributions of Sara Tonelli and Alessio Scussel in the development of PreMOn.

A PreMOn Ontology Resource Specific Modules

A.1 PreMOn Ontology – PropBank

We define classes pmopb:Roleset and pmopb:SemanticRole as subclasses of pmo:SemanticClass and pmo:SemanticRole, respectively. Each pmopb:SemanticRole instance is related (via property pmopb:argument) to exactly one pmopb:Argument, which is defined as the disjoint union of three subclasses: pmopb:NumberedArgument, containing the individuals corresponding to numbered arguments (e.g., Arg0, Arg1); pmopb:Modifier, containing the individuals corresponding to modifiers (e.g., ArgM-LOC, ArgM-TMP); and, pmopb:SecondaryAgent, containing the single individual annotating secondary agents (ArgA). While PB annotation guidelines define a single modifier (ArgM) with multiple function tags (e.g., LOC, TMP), we opt to specialize the modifier for each function tag, similarly to the way these arguments are actually annotated by state-of-the-art SRL tools. Property pmopb:tag enables associating possible tags, either a pmopb:Modifier or some additional tag defined in class pmopb:Tag, to pmopb:SemanticRoles, or nif:Annotations of semantic roles in examples. Additional classes (and related properties) are defined to represent inflectional information about examples: pmopb:Inflection, pmopb:Person, pmopb:Tense, pmopb:Aspect, pmopb:Voice, and pmopb:Form.

30 While this class does not necessarily specialize pmo:SemanticRole with additional properties or restrictions, we add it to ease the retrieval of PB-specific semantic roles, something handy when the same repository contains roles from several predicate models.
A.2 PreMOn Ontology – NomBank

Similarly to PB, we define \textit{pmonb:RoleSet} and \textit{pmonb:SemanticRole} as subclasses of \textit{pmo:SemanticClass} and \textit{pmo:SemanticRole}, respectively. Each \textit{pmonb:SemanticRole} instance is related (via property \textit{pmonb:argument}) to exactly one \textit{pmonb:Argument}, which is defined as the disjoint union of two subclasses: \textit{pmonb:NumberedArgument}, containing the individuals corresponding to numbered argument (e.g., Arg0, Arg1), and \textit{pmonb:Modifier}, containing the individuals corresponding to modifiers (e.g., ArgM-LOC, ArgM-TMP). We also define class \textit{pmonb:Tag} to capture (via property \textit{pmonb:tag}) some specific annotations of markables (e.g., PRD, REF, SUPPORT) in the examples.

A.3 PreMOn Ontology – VerbNet

We define classes \textit{pmovn:VerbClass} and \textit{pmovn:SemanticRole} as subclasses of \textit{pmo:SemanticClass} and \textit{pmo:SemanticRole}, respectively. VN class members are modeled as instances of \textit{ontolex:LexicalEntry}, connected to their class via property \textit{ontolex:evokes}. The VN class hierarchy is modeled via the \textit{pmovn:subclassOf} property (subproperty of \textit{skos:broader}), that relates a verb class (e.g., 13.1-1) with its parent class (e.g., 13.1) —see Figure 8 for an instantiation of this property. Given the propagation of semantic roles along the class hierarchy, we introduce property \textit{pmovn:definesSemRole} to differentiate the \textit{pmovn:SemanticRole} instances defined on a class from the ones inherited from its ancestor classes. Each \textit{pmovn:SemanticRole} instance is related (via property \textit{pmovn:thematicRole}) to exactly one \textit{pmovn:ThematicRole}, which contains all the thematic roles defined in VN. These thematic roles are organized in a hierarchy, which is formalized via the \textit{skos:broader} property. For instance,

\texttt{pmovn:agent skos:broader pmovn:actor}

states that \textit{pmovn:agent} is more specific than \textit{pmovn:actor}. VN selectional restrictions on \textit{pmovn:SemanticRoles} (e.g., restricting “theme” to something not animate) are formalized using property \textit{pmovn:restriction} and class \textit{pmovn:Restriction}. A verb class may have one or more

\footnote{Further details on the formalization of selectional / syntactic restrictions are reported on PreMOn website.}
pmovn:VerbNetFrames (via property pmovn:frame, or its subproperty pmovn:definesFrame, to distinguish frames defined on the class or inherited from ancestors), which have one or more ordered pmovn:SynItems modeling a syntactic construction (e.g., “Agent V Theme [sentential]”) shared by all members of the class, and one or more ordered semantic pmovn:Preds, modeling the meaning of the event, and its participants, expressed by the verb class for that syntactic construction (e.g., “approve(during(E), Agent, Theme)”). pmovn:SynItems are specialized according to their syntactical function (e.g., pmovn:NpSynItem for noun phrases). A pmovn:NpSynItem can point (via pmovn:valueObj) to a pmovn:SemanticRole, and define, via pmovn:restriction, (i) a selectional restriction holding for the pmovn:SemanticRole in that frame (e.g., “animate”), or (ii) some other syntactic restriction (e.g., “np to inf”). Similarly, selectional restrictions can be modelled on pmovn:PrepSynItem (e.g., “spatial”). Predicates in pmovn:Pred have a type (pmovn:PredType, e.g., “approve”) and can be further decomposed in pmovn:PredArg (e.g., “during(E)”) of various types (e.g., pmovn:EventPredArg). Negation of a predicate is expressed by typing the corresponding instance as pmovn:NegPred, while implicit pmovn:PredArg are typed as pmovn:ImplicitArg.

A.4 PreMOn Ontology – FrameNet

We define classes pmofn:Frame and pmofn:FrameElement as subclasses of pmo:SemanticClass and pmo:SemanticRole, respectively. pmofn:FrameElement is further specialized in four subclasses, denoting the four typologies of FN frame elements (e.g., pmofn:CoreFrameElement). Being pmo:SemanticRoles, in PreMOn Ontology frame elements are always specific to the frame where they are defined, even for extra thematic frame elements that are typically shared across frames in FN (e.g., the “Circumstances” extra thematic frame element corresponds to multiple individuals of type pmofn:ExtraThematicFrameElement, one for each frame where it is defined). Frame element core sets of a pmofn:Frame are represented as reified objects of type pmofn:FECoreSet, having as members some pmofn:FrameElements. Relations between pmofn:Frames are modeled using the subproperties of pmofn:frameRelation (e.g., pmofn:inheritsFrom). Similarly, mappings between pmofn:FrameElements of pmofn:Frames related via some pmofn:frameRelation are represented using frame relation-specific subproperties of pmofn:frameElementRelation (e.g., pmofn:inheritsFromFER). Within a frame, a frame element may exclude/require the presence of another frame element (pmofn:excludesFrameElement/pmofn:requiresFrameElement). pmofn:LexicalUnit, associating a lexical entry with a frame, is defined as subclass of pmofn:Conceptualization. A pmofn:LexicalUnit may have a development status (pmofn:LUStatus) and can incorporate a pmofn:FrameElement (e.g., “microwave”, besides evoking frame “Apply heat”, also incorporates the frame element “Heating instrument”). Finally, pmofn:Frames, pmofn:FrameElements and pmofn:LexicalUnits can be constrained according to some semantic types, defined in pmofn:SemType, and organized in a hierarchy according to pmofn:subTypeOf relations between them.

32 We relied on the standard first/next/item pattern for lists.
References


