Verifying Integrity Constraints of a RDF-based WordNet

Fabricio Chalub[1] and **Alexandre Rademaker**[1]

[1]IBM Research Avenida Pasteur, 138. Rio de Janeiro, Brazil

alexrad@br.ibm.com,fchalub@br.ibm.com

Abstract

This paper presents our first attempt at verifying integrity constraints of our openWordnet-PT against the ontology for Wordnets encoding. Our wordnet is distributed in Resource Description Format (RDF) and we want to guarantee not only the syntax correctness but also its semantics soundness.

1 Introduction

Lexical databases are organized knowledge bases of information about words. These resources typically include information about the possible meanings of words, relations between these meanings, definitions and phrases that exemplify their use and maybe some numeric grades of confidence in the information provided. The Princeton English Wordnet (Fellbaum, 1998), is probably the most popular model of a lexical knowledge base. Our main goal is to provide good quality lexical resources for Portuguese, making use, as much as possible, of the effort already spent creating similar resources for English. Thus we are working towards a Portuguese wordnet, based on the Princeton model (de Paiva et al., 2012).

In a previous paper (Real et al., 2015) we reported the new web interface[1] for searching, browsing and collaborating on the improvement of OpenWordnet-PT. Correcting and improving linguistic data is a hard task, as the guidelines for what to aim for are not set in stone nor really known in advance. While the WordNet model has been paradigmatic in modern computational lexicography, this model is not without its failings and shortcomings, as far as specific tasks are concerned. Also it is easy and somewhat satisfying to provide copious quantitative descriptions of numbers of synsets, for different parts-of-speech, of triples associated to these synsets and of intersections with different subsets of Wordnet, etc. However, the whole community dedicated to creating wordnets in other languages, the Global WordNet Association[2], has not come up with criteria for semantic evaluation of these resources nor has it produced, so far, ways of comparing their relative quality or accuracy. Thus qualitative assessment of a new wordnet seems, presently, a matter of judgment and art, more than a commonly agreed practice.

Believing that this qualitative assessment is important, and so far rather elusive, we propose that having many eyes over the resource, with the ability to shape it in the directions wanted, is a main advantage. This notion of volunteer curated content, as first and foremost exemplified by Wikipedia, needs adaptation to work for lexical resources.

Our openWordnet-PT was distributed since its beginning in RDF, following the Semantic Web standards proposed by Tim Berners-Lee (Berners-Lee, 1998). Nevertheless, so far, although we make available not only the data but also its model definition in OWL[3], we have not addressed the task to confront the data with its model to guarantee that data is compliance with the defined model. This is the main contribution of this paper.

2 OpenWordnet-PT

The OpenWordnet-PT (Rademaker et al., 2014), abbreviated as OpenWN-PT, is a wordnet originally developed as a projection of the Universal WordNet (UWN) (de Melo and Weikum, 2009). Its long term goal is to serve as the main lexicon for a system of natural language processing focused on logical reasoning, based on representation of knowledge, using an ontology, such as SUMO (Pease and Fellbaum, 2010).

[1] http://wnpt.brlcloud.com/wn/

[2] http://globalwordnet.org/

[3] https://github.com/own-pt/openWordnet-PT

OpenWN-PT has been constantly improved through *linguistically motivated* additions and removals, either manually or by making use of large corpora. This is also the case for the lexicon of nominalizations, called NomLex-PT, that is integrated to the OpenWN-PT (Freitas et al., 2014). One of the features of both resources is to try to incorporate different kinds of quality data already produced and made available for the Portuguese language, independent of which variant of Portuguese one considers.

The philosophy of OpenWN-PT is to maintain a close connection with Princeton's wordnet since this minimizes the impact of lexicographical decisions on the separation or grouping of senses in a given synset. Such disambiguation decisions are inherently arbitrary (Kilgarriff, 1997), thus the multilingual alignment gives us a pragmatic and practical solution. It is practical because Princeton WordNet remains the most used lexical resource in the world. It is also pragmatic, since those decisions will be more useful, if they are similar to what other wordnets say. Of course this does not mean that all decisions will be sorted out for us. As part of our processing is automated and error-prone, we strive to remove the biggest mistakes created by automation, using linguistic skills and tools. In this endeavor we are much helped by the linked data philosophy and implementation, as keeping the alignment between synsets is facilitated by looking at the synsets in several different languages in parallel. For this we make use of the Open Multilingual WordNet's interface (Bond and Foster, 2013) through links from our interface.

This lexical enrichment process of OpenWN-PT reported in employs three language strategies: (1) translation; (2) corpus extraction; and (3) dictionaries. The interested reader will find more details in (Rademaker et al., 2014; Real et al., 2015). The essential fact is that given the constant release of new versions of our openWN-PT, we must ensure the quality of the data that we make available. By quality here we mean not only the data content but its encoding consistency.

3 OpenWordnet-PT in RDF

As reported in (Rademaker et al., 2014), since its beginning OpenWN-PT is distributed using the Resource Description Format (RDF) (Cyganiak and Wood, 2003). We have being following the increasingly popular way of addressing the issue of interoperability by relying on Linked Data and Semantic Web standards such as RDF and OWL (Hitzler et al., 2012), which have led to the emergence of a number of Linked Data projects for lexical resources (de Melo and Weikum, 2008; Chiarcos et al., 2012). The adoption of such standards not only allows us to publish both the data model and the actual data in the same format, they also provide for instant compatibility with a vast range of existing data processing tools and storage systems, triple stores, providing query interfaces based on the SPARQL standard (Harris and Seaborne, 2013).

To encode any data in RDF, one needs to decide which classes and properties (vocabulary) will be used. The adoption of already defined vocabularies helps on the data interoperability since these makes data easily integrate with other resources.

We chose to use the vocabulary for wordnets encoding proposed by (van Assem et al., 2006) which is based on Princeton Wordnet 2.0. Their work includes (1) a mapping of WordNet 2.0 concepts and data model to RDF/OWL; (2) conversion scripts from the WordNet 2.0 Prolog distribution to RDF/OWL files; and (3) the actual WordNet 2.0 data. The suggested representation stayed as close to the original source as possible, that is, it reflects the original WordNet data model without interpretation. The WordNet schema proposed by (van Assem et al., 2006) has three main classes: *Synset*, *WordSense* and *Word*. The first two classes have subclasses for each lexical group present in WordNet. Each instance of Synset, WordSense and Word has its own URI.

Since (van Assem et al., 2006) is based on Princeton Wordnet 2.0, its use required few adaptations. Our first decision was to adapt the WordNet 2.0 vocabulary to version 3.0, having our own URIs for all entities (classes and properties). We converted the WordNet 3.0 data to RDF in such a way that OpenWN-PT is an extension of WordNet 3.0, with its instances, connected to Princeton instances through *owl:sameAs* relations. That is, for each Princeton WordNet synset, we created an equivalent synset in OpenWN-PT synset, with no additional synsets or relations so far. Given that OpenWN-PT's RDF is only useful together with an RDF version of Princeton WordNet and we wanted to ensure that all information in the WordNet 3.0 distribution was transformed to RDF, we wrote our own script to translate the Princeton

WordNet 3.0 data files to RDF so they can be distributed alongside OpenWN-PT.[4].

For the URI schema, we adopted a similar approach of (van Assem et al., 2006) of pattern for the URIs by classes. Moreover, we created the domain `https://w3id.org/own-pt/` under our control as suggested by the Linked Data principles. In Table 1, under the namespace [1] we have the classes and properties of our vocabulary (TBox), adapted from (van Assem et al., 2006). The namespace [2] holds the instances of our openWordnet-PT and [3] holds the Princeton instances. Our Nomlex-PT (Freitas et al., 2014) data also has its vocabulary and data namespace, respectively, [4] and [5].

```
1  https://w3id.org/own-pt/wn30/schema/
2  https://w3id.org/own-pt/wn30-pt/instances/
3  https://w3id.org/own-pt/wn30-en/instances/
4  https://w3id.org/own-pt/nomlex/schema/
5  https://w3id.org/own-pt/nomlex/instances/
```

Table 1: the used URIs

4 Consistency check of OWL and Integrity Constraints in RDF

The Web Ontology Language (OWL) [5] is a family of knowledge representation languages for authoring ontologies (or Knowledge bases) composed by OWL Lite, OWL DL and OWL Full. The OWL languages are built upon the W3C standard RDF and characterized by formal semantics. OWL Lite and OWL DL semantics are based on Description logics (DLs) (Baader, 2003). DL are a family of logics that are decidable fragments of first-order logic with attractive and well-understood computational properties.

A DL knowledge base is comprised by two components, TBox and ABox. The TBox contains intensional knowledge in the form of a terminology and is built through declarations of the general properties of concepts[6]. The ABox contains extensional knowledge, also called assertional knowledge. The knowledge that is specific to the individuals of the domain of discourse. Intensional knowledge is usually thought not to change and extensional knowledge is usually thought to be contingent, and therefore subject to occasional or even constant change.

Given an ontology encoded in OWL (Lite or DL) one can use DL reasoners for different tasks such as: concepts consistency checking, query answering, classification, etc. In particular, classification amounts to placing a new concept expression in the proper place in a taxonomic hierarchy of concepts, it can be accomplished by verifying the subsumption relation between each defined concept in the hierarchy and the new concept expression. Validating an ontology means to guarantee that all concepts are satisfiable, that is, the concepts definition do not contain contradictions.

The basic reasoning task in an ABox is instance checking, which verifies whether a given individual is an instance of (or belongs to) a specified concept. Although other reasoning services are usually employed, they can be defined in terms of instance checking. Among them we find knowledge base consistency, which amounts to verifying whether every concept in the knowledge base admits at least one individual; realization, which finds the most specific concept an individual object is an instance of; and retrieval, which finds the individuals in the knowledge base that are instances of a given concept (query answering).

In some use cases, we need a method to validating the RDF data regarding a given model. In this case, OWL users intend OWL axioms to be interpreted as constraints on RDF data (Pérez-Urbina et al., 2012). For that, one has to define a semantics for OWL based on the Closed World Assumption and a weak variant of the Unique Name Assumption (Baader, 2003). OWL default semantics adopts the Open World Assumption (OWA) and does not adopt the Unique Name Assumption (UNA). These design choices make it very difficult to treat these axioms as ICs. On the one hand, due to OWA, a statement must not be inferred to be false on the basis of failures to prove it; therefore, the fact that a piece of information has not been specified does not mean that such information does not exist. On the other hand, the absence of UNA allows two different constants to refer to the same individual.

In the next section, we present some preliminary experiments with TBox and ABox consistency check and integrity constraints (IC) validation in our RDF/OWL data, reporting our experience with most well-know freely available tools. Nevertheless, it is important to emphasize the capabilities that semantic web technologies that ex-

[4] `https://github.com/own-pt/wordnet2rdf`

[5] `http://www.w3.org/OWL/`

[6] In this paper the TBox is sometimes called the vocabulary.

ceed the currently mainstream technologies.

Most research groups that are still using XML for lexical resources distribution would argue that XML Schema (Fallside and Walmsley, 2004) can ensure some constraints that we verify in the next section. Relational database users would argue that SQL is an already mature and declarative query language. We argue that OWL/RDF brings much more expressivity allowing much more robust and semantics aware verification with queries such as:

```
select ?w ?ws1 ?ws2
{
  ?ss1 wn30:containsWordSense ?ws1 .
  ?ws1 wn30:word ?w .
  ?ss2 wn30:containsWordSense ?ws2 .
  ?ws2 wn30:word ?w .
  ?ss1 wn30:hyponymOf* ?ss2 .
}
```

In the SPARQL query above, we are asking for words that occur repeated in the same branch of the hierarchy of synsets formed by the `wn30:hyponymOf` transitive closure.

5 Validating OpenWN-PT

We were interested in checking our RDF and OWL files against a wide variety of errors, both minor and major and to increase our coverage we opted to use a variety of reasoners.

We started with Protégé [7], which is an ontology editor that among other features has interface with two well-know DL reasoners: FaCT++ (Tsarkov and Horrocks, 2006) and HermiT (Shearer et al., 2008). Starting in version 4, Protégé also gives us the opportunity to search for explanations that caused an inconsistency (Horridge et al., 2008). Racer (Haarslev et al., 2012) and Pellet (Sirin et al., 2007) are reasoners that have this feature built-in.

In order to verify OWN-PT files we needed to combine all files in `https://github.com/own-pt/openWordnet-PT` and the Simple Knowledge Organization System (SKOS) [8] ontology file. There are a number of tools available for this, we chose *RDF$_{pro}$* (Corcoglioniti et al., 2015), which was the fastest in our benchmarks.

The errors found can be categorized in three different classes: datatype errors, domain and range errors, structural errors.

5.1 Datatype errors

Errors such as missing datatype declarations and wrongly typed literals were found by both Hermit and Pellet. Hermit identified the following missing classes:

```
wn30:AdjectiveWordSense
   rdfs:subClassOf wn30:WordSense .

wn30:VerbWordSense
   rdfs:subClassOf wn30:WordSense .
```

And the following verification fails due to incorrectly typed literals:

```
Literal value "00113726" does not
  belong to datatype nonNegativeInteger

Literal value "104" does not belong
  to datatype nonNegativeInteger
```

These errors were caused by the fact that `wn30:synsetId` and `wn30:tagCount` are defined as properties of synsets and word senses that are non-negative integers, but they were incorrectly stored without the type qualifier, for example: the literal in `synset-13363970-n synsetId "13363970"` should have been specified as `"13363970"^^xsd:nonNegativeInteger`.

Pellet Lint, like lint tools for programming languages, aims to detect possibly incorrect constructions that generally indicate bugs. For brevity we omit the prefix `https://w3id.org/own-pt/` from the individuals below.

```
[Untyped classes]
wn30/schema/BaseConcept
nomlex/schema/Nominalization
wn30/schema/CoreConcept
[...]
```

```
[Untyped datatype properties]
wn30/schema/senseKey
wn30/schema/syntacticMarker
wn30/schema/lexicographerFile
[...]
```

```
[Untyped individuals]
wn30-en/instances/wordsense-01362387-a-2
wn30-en/instances/wordsense-01362387-a-1
wn30-en/instances/wordsense-01722140-a-1
[...]
```

What Pellet Lint calls an untyped class is an object of a triple involving `rdf:type`, but it was never formally defined as an OWL class. The same idea applies to untyped properties: these are never formally defined as an OWL property,

[7] `http://protege.stanford.edu/`

[8] `http://www.w3.org/TR/skos-reference/`

and lacks any information about its domain and range. Untyped individuals also are used as objects, but never participate in triples as a subject, which seems like a mistake on some previous data import task. These likely need to be removed.

5.2 Domain and range errors

Moving beyond these initial type checks, we used initially Protégé with the FaCT++ reasoner to match our triple store statements against the OWL definition. The ontology was found to be inconsistent, with the following explanation:

```
Explanation for: Thing SubClassOf Nothing
classifiedByRegion Domain Synset
current_account classifiedByRegion Britain
current_account Type WordSense
Synset DisjointWith WordSense
```

We now give a detailed analysis of this explanation; we'll omit such details from the other inconsistencies found later on this section. The relation `wn30:classifiedByRegion` was created from the `;r` pointer symbol in Princeton WordNet data distribution, documented in `wninput(5wn)`.[9] In the explanation above, `current_account` is the label of `wordsense-13363970-n-3` and `Britain` the label of `wordsense-08860123-n-4`. These two subjects are related via the following triple:

```
wordsense-13363970-n-3 classifiedByRegion
    wordsense-08860123-n-4
```

This triple was generated from the following line in original Princeton `data.noun` file (formatted for clarity):

```
13363970 21 n 03
  checking_account 0 chequing_account 0
  current_account 1 004
   @ 13359690 n 0000
  ;r 08860123 n 0304
  ;r 08820121 n 0201
  ;r 09044862 n 0101
  | a bank account against which the
    depositor can draw checks that are
    payable on demand
```

Notice that the triple in the explanation above is a relationship between two word senses, while our definition of the `wn30:classifiedByRegion` property is as follows:

```
wn30:classifiedByRegion
  a rdf:Property, owl:ObjectProperty ;
  rdfs:domain wn30:Synset ;
  rdfs:range wn30:NounSynset ;
  rdfs:subPropertyOf wn30:classifiedBy .
```

In other words, it is a property whose domain contains synsets and its range contains all noun synsets. This is contradicted by the example, where the `rdfs:domain` and `rdfs:range` restrictions were violated.

To fix the inconsistency, we need to understand the source of the error: is the problem in our translation from the Wordnet file to RDF, the OWL definition of `wn30:classifiedByRegion`, or an issue in Wordnet itself? In the excerpt from `data.noun` above, all three domain/region pointers are between word senses, which was preserved in the translation to RDF. Looking at the other entries there, we find that `chequing_account` and `Canada` and `checking_account` and `United_States` are also word senses labels that are related by `wn30:classifiedByRegion`. This indicates a desire to differentiate between the different lexical forms and their regions of usage, which can be seen as a form of lexical relationship. This indicates an issue with the formalization of the relation `wn30:classifiedByRegion`. Going back to the original definition in `wninput(5wn)` we find the following (emphasis ours):

> The following pointer types are *usually* used to indicate lexical relations: Antonym, Pertainym, Participle, Also See, Derivationally Related. The remaining pointer types are *generally* used to represent semantic relations.

While generally a domain/region pointer is a semantic relationship, our examples show that this is not always the case. Also, by using words such as 'generally' and 'usually' the informal description above accommodates such cases. This leads us to think that `wn30:classifiedByRegion` is both a semantic and a lexical relation, unlike our formal definition states.

We can query for the statistics of the `wn30:classifiedByRegion` domain in our endpoint.[10] The SPARQL query below selects all individuals that are involved in `wn30:classifiedByRegion` relations, their

[9] `http://goo.gl/AbkdaZ`

[10] `http://goo.gl/ptPw6S`

types, and counts the number of individual by type.

```
select ?t (count(?t) as ?ct)
{ ?s wn30:classifiedByRegion ?o ;
     a ?t
} group by ?t
```

The majority of the subjects – over 1200 – are synsets, but there are 15 word senses as well, meaning that wn30:classifiedByRegion is definitely not strictly a semantic relation. To fix this issue, the definition needed to be changed so that the domain and range contains both synsets and word senses. This is done using the owl:unionOf operator, which represents set union.

```
wn30:classifiedByRegion
  a rdf:Property, owl:ObjectProperty ;
  rdfs:subPropertyOf wn30:classifiedBy ;
  rdfs:range [ a owl:Class ;
    owl:unionOf (wn30:NounWordSense
                 wn30:NounSynset)] ;
  rdfs:domain [ a owl:Class ;
    owl:unionOf (wn30:WordSense wn30:Synset)] .
```

We found similar problems with the properties wn30:frame, wn30:classifiedByUsage and wn30:classifiedByTopic. We selected the latter since it highlights one of the issues that we find while performing formal verifications, which is the complexity of the proofs/explanations. This is the explanation found for the issue:

```
synset-01345109-v hypernymOf
    synset-01220528-v
 VerbWordSense subClassOf WordSense
 frame domain VerbWordSense
 synset-01220528-v frame
    "Somebody ----s something"
 hypernymOf range Synset
 Synset disjointWith WordSense
```

While this example can be understood, it definitely could be made simpler. For instance, synset-01220528-v found to be of type 'synset' due to the fact that it is the object of a triple containing the predicate wn30:hypernymOf combined with that fact that the range of this predicate is the set of all synsets. A more concise way is to realize that synset-01220528-v is a verb synset and that verb synsets are a subset of synsets. In any case, interpreting the explanation, we see that wn30:frame is being used as a relation whose domain contains a synset, but its definition prohibits this. We can query our triple store for the *de facto* domains of wn30:frame via a SPARQL query similar to the one used for wn30:classifiedByRegion. We again omit the results for brevity, but there are both word senses and synsets in the domain of this relation. Checking the definition of wn30:frame in wninput(5wn) we find that its original formal definition is too restrictive as it allows frames to exist between both synsets *and* word senses.

After fixing those, only a couple of issues remained:

```
NounSynset SubClassOf Synset
hemolysis Type WordSense
holonymOf Domain NounSynset
adjectivePertainsTo SubPropertyOf meronymOf
meronymOf SubPropertyOf inverse(holonymOf)
Synset DisjointWith WordSense
haemolytic adjectivePertainsTo hemolysis
```

While wn30:adjectivePertainsTo is a relation between word senses, it was marked as a subproperty of wn30:meronymOf, which is a relationship between synsets. It was also marked as the inverse of wn30:holonymOf, which is also a semantic relation. Both restrictions are, of course, incorrect and were removed.

The final issues were investigated using the Pellet reasoner. This allows us to verify our work and also experiment with the different implementations of the explanations for inconsistencies.

```
Axiom: Thing subClassOf Nothing

inSynset range Synset
VerbWordSense subClassOf WordSense
synset-00105023-a containsWordSense
  wordsense-00105023-a-2
synset-00105023-a seeAlso synset-00885415-a
AdjectiveWordSense subClassOf WordSense
seeAlso domain AdjectiveWordSense
            or VerbWordSense
inSynset inverseOf containsWordSense
Synset disjointWith WordSense
```

Here, wn30:seeAlso usually indicates lexical relations, but the explanation shows relationship between two synsets.

5.3 Structural errors

Our last example show cases yet another trap that should be avoided when designing ontologies, which is to assume that once it is consistent, there is nothing else to do. In our case, our modifications so far lead us to a consistent ontology, but unfortunately that doesn't mean that there weren't any issues left. In fact, there were two extremely serious errors in our RDF distribution that were not caught by the analyses so far and were found accidentally through a cursory look: during one of our post-processing jobs we mistakenly implemented a blank node renaming algorithm and ended up having two invalid situations: (a) two or more words associated to a single word sense subject; (b) two or more lexical forms associated to a single word subject.

After fixing our ontology to give the proper restrictions on word senses, words, and lexical forms, Pellet was able to identify the issues. The following excerpt describes a single word sense (`wordsense-01860795-v-2`) with two words associated ('deixar', 'parar').

```
wordsense-01860795-v-2 type WordSense
word-deixar lexicalForm "deixar"@pt
word-parar lexicalForm "parar"@pt
wordsense-01860795-v-2 word word-deixar
Word subClassOf lexicalForm exactly 1
wordsense-01860795-v-2 word word-parar
word-deixar type Word
word-parar type Word
WordSense subClassOf word exactly 1 Word
```

The last tool that we tested was Stardog [11]. Stardog is the only reasoner and database system that supports ICV. Under the ICV semantics, the axioms below from the `wn30:WordSense` class were taken as constraints rather than terminology definitions. In other words, if Stardog finds an instance of the class `wn30:WordSense` connected to more than one instance of `wn30:Word`, it will raise an exception instead of infer that the two different `wn30:Word` instances should be the same.

```
wn30:WordSense
   a rdfs:Class, owl:Class ;
   rdfs:subClassOf [
    a owl:Restriction ;
    owl:onProperty wn30:inSynset ;
    owl:qualifiedCardinality
      "1"^^xsd:nonNegativeInteger ;
     owl:onClass wn30:Synset ], [
    a owl:Restriction ;
    owl:onProperty wn30:word;
    owl:qualifiedCardinality
     "1"^^xsd:nonNegativeInteger ;
    owl:onClass wn30:Word ] .
```

Unfortunately, in all tests that we run, Stardog hung without producing any output, even when we executed it with few axioms of our ontology. We hope to investigate the problem in a future report.

6 Conclusion

The use of different systems, with different functionalities, give us more confidence in our validations. Unfortunately, it required considerable effort to prepare data in different formats and interpret the results. Racer and RDFUnit did not give us meaningful results. We could not use Stardog at all. We will continue to try them, though, as we believe the diversity of tools and techniques are beneficial to the coverage of potential problems.

Performance is still an issue. Some of these experiment took hours to complete, in a relatively simple ontology. It looks like most of DL reasoners are not prepared to handle large ABoxes.

Most DL reasoners are based on some variation of tableaux or other refutation based procedure (Baader, 2003). Prove by refutation does not preserve information and tableaux proofs usually have exponential size. In the future, we hope to implement a proof-theoretical based reasoner for DL based on (Rademaker, 2012).

It is also worthy to mention that the tools that we tested do not always have an user-friendly interface, making adoption for people outside the area difficult.

Reasoning with closed world assumption for ICV is a future work given the problems that we faced with Stardog. Finally, DL Learning (Lehmann, 2009) and Shapes Constraint Language (Knublauch and Ryman, 2016) are another possible interesting techniques to explorer. The former would allow us to extract the minimum required TBox for a given ABox, the latter would be an alternative language for expressing constraints.

Aknowledgements This work used Protégé resource, which is supported by grant GM10331601 from the National Institute of General Medical Sciences of the United States National Institutes of Health.

References

[Baader2003] Franz Baader. 2003. *The description logic handbook: theory, implementation, and applications*. Cambridge university press.

[Berners-Lee1998] Tim Berners-Lee. 1998. Semantic web road map. Technical report, W3C, September.

[Bond and Foster2013] Francis Bond and Ryan Foster. 2013. Linking and extending an open multilingual wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, Sofia, Bulgaria, August. Association for Computational Linguistics.

[Chiarcos et al.2012] Christian Chiarcos, Sebastian Nordhoff, and Sebastian Hellmann. 2012. *Linked*

[11] `http://www.stardog.com`.

data in linguistics: Representing and connecting language data and language metadata. Springer.

[Corcoglioniti et al.2015] Francesco Corcoglioniti, Marco Rospocher, Michele Mostarda, and Marco Amadori. 2015. Processing billions of rdf triples on a single machine using streaming and sorting. In *ACM SAC 2015 Proceedings*.

[Cyganiak and Wood2003] Richard Cyganiak and David Wood. 2003. RDF 1.1 concepts and abstract syntax. Technical Report Draft 23 July 2013, W3C.

[de Melo and Weikum2008] Gerard de Melo and Gerhard Weikum. 2008. Language as a foundation of the Semantic Web. In *Proc. of ISWC 2008*, volume 401.

[de Melo and Weikum2009] Gerard de Melo and Gerhard Weikum. 2009. Towards a universal wordnet by learning from combined evidence. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009)*, pages 513–522, New York, NY, USA. ACM.

[de Paiva et al.2012] Valeria de Paiva, Alexandre Rademaker, and Gerard de Melo. 2012. OpenWordNet-PT: An open Brazilian wordnet for reasoning. In *Proceedings of 24th International Conference on Computational Linguistics*, COLING (Demo Paper).

[Fallside and Walmsley2004] David C. Fallside and Priscilla Walmsley. 2004. Xml schema part 0: primer second edition. Technical Report W3C Recommendation 28 October 2004, W3C.

[Fellbaum1998] Christiane Fellbaum, editor. 1998. *WordNet: An electronic lexical database*. MIT Press.

[Freitas et al.2014] Cláudia Freitas, Valeria de Paiva, Alexandre Rademaker, Gerard de Melo, Livy Real, and Anne de Araujo Correia da Silva. 2014. Extending a lexicon of portuguese nominalizations with data from corpora. In Jorge Baptista, Nuno Mamede, Sara Candeias, Ivandré Paraboni, Thiago A. S. Pardo, and Maria das Graças Volpe Nunes, editors, *Computational Processing of the Portuguese Language, 11th International Conference, PROPOR 2014*, São Carlos, Brazil, oct. Springer.

[Haarslev et al.2012] Volker Haarslev, Kay Hidde, Ralf Möller, and Michael Wessel. 2012. The RacerPro knowledge representation and reasoning system. *Semantic Web Journal*, 3(3):267–277.

[Harris and Seaborne2013] Steve Harris and Andy Seaborne. 2013. SPARQL 1.1 query language. Technical Report W3C Recommendation 21 March 2013, W3C.

[Hitzler et al.2012] Pascal Hitzler, Markus Krotzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph. 2012. OWL 2 web ontology language primer. Technical Report W3C Rec 11 Dec 2012, W3C.

[Horridge et al.2008] Matthew Horridge, Bijan Parsia, and Ulrike Sattler. 2008. Explanation of OWL entailments in protégé 4. In *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC2008), Karlsruhe, Germany, October 28, 2008.*

[Kilgarriff1997] Adam Kilgarriff. 1997. I don't believe in word senses. *Computers and the Humanities*, 31(2):91–113.

[Knublauch and Ryman2016] Holger Knublauch and Arthur Ryman. 2016. Shapes constraint language (shacl). Technical Report W3C Working Draft 28 January 2016, W3C. `http://www.w3.org/TR/shacl/`.

[Lehmann2009] Jens Lehmann. 2009. DL-Learner: learning concepts in description logics. *Journal of Machine Learning Research (JMLR)*, 10:2639–2642.

[Pease and Fellbaum2010] Adam Pease and Christiane Fellbaum. 2010. Formal ontology as interlingua: the SUMO and WordNet linking project and global WordNet linking project. In *Ontology and the Lexicon: A Natural Language Processing Perspective*, Studies in Natural Language Processing, chapter 2, pages 25–35. Cambridge University Press.

[Pérez-Urbina et al.2012] Héctor Pérez-Urbina, Evren Sirin, and Kendall Clark. 2012. Validating rdf with owl integrity constraints. Technical report, Clark & Parsia, LLC.

[Rademaker et al.2014] Alexandre Rademaker, Valeria de Paiva, Gerard de Melo, Livy Maria Real Coelho, and Maira Gatti. 2014. Openwordnet-pt: A project report. In Heili Orav, Christiane Fellbaum, and Piek Vossen, editors, *Proceedings of the 7th Global WordNet Conference*, Tartu, Estonia, jan.

[Rademaker2012] Alexandre Rademaker. 2012. *A Proof Theory for Description Logics*. SpringerBriefs in Computer Science. Springer.

[Real et al.2015] Livy Real, Fabricio Chalub, Valeria dePaiva, Claudia Freitas, and Alexandre Rademaker. 2015. Seeing is correcting: curating lexical resources using social interfaces. In *Proceedings of the 4th Workshop on Linked Data in Linguistics: Resources and Applications*, pages 20–29, Beijing, China, July. Association for Computational Linguistics.

[Shearer et al.2008] R. Shearer, B. Motik, and I. Horrocks. 2008. HermiT: a highly efficient OWL reasoner. In *Proceedings of the Fifth International Workshop on OWL: Experiences and Directions (OWLED)*.

[Sirin et al.2007] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. 2007. Pellet: A practical OWL-DL reasoner. *Web Semant.*, 5(2):51–53, June.

[Tsarkov and Horrocks2006] Dmitry Tsarkov and Ian Horrocks. 2006. FaCT++ description logic reasoner: System description. In *Proceedings of the Third International Joint Conference on Automated Reasoning*, IJCAR'06, pages 292–297, Berlin, Heidelberg. Springer-Verlag.

[van Assem et al.2006] Mark van Assem, Aldo Gangemi, and Guus Schreiber. 2006. RDF/OWL representation of WordNet. Technical Report W3C Working Draft 19 June 2006, W3C.