# Extending a Wordnet Framework for Simplicity and Scalability

**Pedro Fialho, Sérgio Curto, Ana Cristina Mendes, Luísa Coheur**

Spoken Language Systems Lab, INESC-ID Lisboa
Instituto Superior Técnico, Technical University of Lisbon
R. Alves Redol, 9 - 2º - 1000-029 Lisboa, Portugal
{pedro.fialho, sergio.curto, ana.mendes, luisa.coheur}@l2f.inesc-id.pt

**Abstract**

The WordNet knowledge model is currently implemented in multiple software frameworks providing procedural access to language instances of it. Frameworks tend to be focused on structural/design aspects of the model thus describing low level interfaces for linguistic knowledge retrieval. Typically the only high level feature directly accessible is word lookup while traversal of semantic relations leads to verbose/complex combinations of data structures, pointers and indexes which are irrelevant in an NLP context. Here is described an extension to the JWNL framework that hides technical requirements of access to WordNet features with an essentially word/sense based API applying terminology from the official online interface. This high level API is applied to the original English version of WordNet and to an SQL based Portuguese lexicon, translated into a WordNet based representation usable by JWNL.

**Keywords:** WordNet, SQL lexicon, abstraction layer

## 1. Introduction

Linguistic thesaurus and dictionaries are useful in Natural Language Processing (NLP) tasks that require semantic enrichment of words or collocations. WordNet (Miller, 1995) provides thesaurus and dictionary functionality through a knowledge model freely available online and offline for open domain in English. Alternative models exist[1] although limited to manual/online access therefore only suitable for knowledge visualization.

WordNet's coverage and detail are appropriate for several NLP tasks (Peh and Ng, 1997), like document clustering (Hotho et al., 2003) and text retrieval (Gonzalo et al., 1998), although widespread usage is best shown with the unofficial adaptations to European languages (Pianta et al., 2002; Vossen, 1997) (among others[2]), closed domains (Bentivogli et al., 2004a; Buscaldi and Rosso, 2008; He, 2006), standard knowledge representations (van Assem and Schreiber, 2006) and fine grained versions (Mihalcea and Moldovan, 2001).

Computational access to the offline/filesystem version is available in several frameworks[3] mostly aimed at representing WordNet's model, therefore lacking procedural word based usage as manually available in the online version[4]. The Java WordNet Library (JWNL)[5] is one such framework and, to our best knowledge, the only featuring translation of filesystem versions to other storage systems. This allows WordNet usage on environments unsuitable for filesystems, such as distributed access to a central version or stateless/diskless systems.

However, JWNL access to WordNet is not simple, making desirable a black-box approach to features already reachable in a too verbose/complex manner, irrelevant for NLP. This was the main motivation for us to develop an NLP component, called JWNLSIMPLE, using linguistic information from WordNet with JWNL as access framework and development starting point, which is here described.

Some of the advantages of JWNLSIMPLE are: most procedures only require a string/sense; only basic WordNet information containers are used; stemming (and some word lookup) procedures cover all Part-of-Speech (PoS); relationship getters (some previously unavailable) map the online WordNet behavior; and the ability to represent/access SQL lexicons as a WordNet (here applied to a Portuguese lexical database). This work contributes by presenting JWNLSIMPLE to the research community, which is available[6] with a technical report[7].

The remainder of this paper is organized as follows: Section 2. briefly describes starting point concepts and components; Section 3. details changes in the framework and intended usage; Section 4. refers subjective value and future work.

## 2. Development baseline

The development starting point of JWNLSIMPLE is focused on components of WordNet 3.0 and JWNL 1.4.1.

### 2.1. WordNet

WordNet is a directed acyclic graph composed of word forms and meanings related in a many to many manner. As thesaurus WordNet describes synonym lemmas in synsets interlinked by lexical and semantic relations, containing typical dictionary information (gloss, PoS and usage examples) and uniquely identified.

The offline WordNet version consists of multiple files with a specific/textual codification while the online version has a typical search engine interface and is the reference WordNet knowledge visualization tool. Non regular inflections (such as "go" and "went") are mapped to lemmas in exception list files (Fellbaum, 1998) while affix based inflec-

[1] http://www.lexipedia.com/
[2] http://www.globalwordnet.org/gwa/wordnet_table.htm
[3] http://wordnet.princeton.edu/wordnet/related-projects/#local
[4] http://wordnetweb.princeton.edu/perl/webwn
[5] http://sourceforge.net/projects/jwordnet/
[6] https://qa.l2f.inesc-id.pt/wiki/index.php/Resources
[7] http://www.inesc-id.pt/ficheiros/publicacoes/8113.pdf

tions are not represented, although described in Morphy[8], the WordNet's original morphological processing library.

### 2.2. JWNL

Usage of JWNLSIMPLE is backed by essential JWNL components. This description complements available manuals[9], API[10] and related works (Maria Teresa Pazienza and Tudorache, 2008b; Cunningham et al., 2011).

**Storage systems** A WordNet filesystem representation may be converted into a serialized object or SQL tables/views defined in a JWNL specific schema. Each storage system corresponds to a concrete implementation of an abstract *Dictionary* managing storage specific resources. WordNet information is accessed through a storage independent API.

**Initialization** An initialization phase sets a WordNet access profile according to an inputted XML specifying storage type, WordNet version/language, allowed morphological operations (language dependent) and storage specific parameters (such as database login or WordNet location).

**Information retrieval** Upon initialization becomes available an instance of *Dictionary* containing word based lookup procedures, most requiring a PoS and returning an implementation specific collection of synsets. Synsets are interlinked by lexical and semantic pointers with properties such as group (lexical/semantic), type, source and targets. It is also possible to find relationships between two synsets.

**Stemming** For stemming purposes, the *Dictionary* provides a *MorphologicalProcessor* allowing broader word coverage by combining affix detachment and exception lists lookup, as seen in Morphy (Fellbaum, 1998). Language specific garbage strings, like detachable suffixes (for gerund and plural resolution) and token delimiters, are defined in JWNL's initialization XML along with lookup operations available on reduced forms.

## 3. Extending JWNL

JWNLSIMPLE works as a Java object requiring a configuration XML (used in JWNL) as argument, allowing multiple WordNet formats/versions to coexist on the same program. Each JWNLSIMPLE should be set and released, imposing a development/usage life cycle. The overall output organization and coverage is based on the official online interface.

### 3.1. Patches

Using database storage, JWNL closes the database connection upon each query execution (as in each word lookup) which is inefficient and error prone on batch usage, thus in JWNLSIMPLE we keep the connection alive until unresponsive or closed explicitly.

### 3.2. New Features

WordNet information containers in JWNL are used as input/output of previously unavailable access features.

**Lookup and Stemming** In JWNLSIMPLE, alternative stemming and word lookup procedures are supplied, most covering all PoS thus providing a larger search space. Stemming procedures wrap the JWNL's stemmer, excluding individual word stems on multiple word expressions ("running away" results only in "run away"), being used internally by the developed lookup procedures.

**Semantic/Linguistic** Getters for relations in Table 1 were obtained by extracting words or synsets related to the input synset with JWNL's (undocumented) pointers guessed by result/example matching with the online WordNet interface.

### 3.3. Wrappers

JWNL getters output implementation specific collections. In JWNLSIMPLE, these are parsed into Java collections.

**List parsers** Getters for relations in Table 2 were based on existent JWNL procedures returning a JWNL specific list representation.

**Tree parsers** Getters for relations in Table 3 were also obtained from existent JWNL procedures instead returning JWNL trees (subgraphs of WordNet for a single relation). JWNLSIMPLE wrappers return a bag of WordNet containers thus discarding tree structure. JWNL trees were traversed with overflow precautions, as such large result sets of inherited/full relations are replaced with their direct/top level equivalents, as seen on the online interface.

### 3.4. SQL based lexicons

JWNLSIMPLE was also applied to a Portuguese WordNet stored in an SQL database according to JWNL's schema. This WordNet contains lexical knowledge from TemaNet[11], Papel[12] and MWN.PT[13] originally represented in a database not compliant with JWNL (dos Santos Correia, 2010). This database was adapted to JWNL's schema with a specific/customizable mapping script since it contains more lexical information than the original WordNet (such as domains and subdomains, concatenated as gloss) and less semantic pointers (only synonym, hypernym or hyponym). Availability of lexical information and semantic pointers is dependent on the originating lexical resource as such the resulting WordNet does not describe the same information slots for all synsets. Some of the information required by JWNL was also not existent in this database thus being filled with nullable characters/numbers.

[8] http://wordnet.princeton.edu/man/morphy.7WN.html
[9] http://sourceforge.net/apps/mediawiki/jwordnet/
[10] http://nlp.stanford.edu/nlp/javadoc/jwnl-docs/
[11] http://www.instituto-camoes.pt/temanet/inicio.html
[12] http://www.linguateca.pt/PAPEL
[13] http://mwnpt.di.fc.ul.pt/features.html

| Relation | Description | Example |
|---|---|---|
| Pertainym | the broad category of a sense | "music" is a pertainym of "musical" |
| Derivationally Related Form | the result of attaching derivational affixes to a stem (a subset of fuzzynyms (Maria Teresa Pazienza and Tudorache, 2008a)) | "fraternity" is a derivationally related form of "fraternal" |
| Phrasal Verb | a multi word expression containing the verb (original meaning may change) | "run away" is a phrasal verb of "run" |
| Domain Usage | the environment/discourse type where a sense is applied (Bentivogli et al., 2004b) | "irony" is a domain usage of "pretty" (as in "pretty mess") |
| Domain Category | the type of domain where a sense is applied | "physics" is a domain category of "light" |
| Instance | a concrete implementation (Miller and Hristea, 2006) (applies to proper nouns) | "Google" is an instance of "search engine" |

Table 1: Some of the covered relations in JWNLSIMPLE without getters in JWNL.

| Relation | Description | Example |
|---|---|---|
| Member Holonym | group where the sense belongs | "forest" is a member holonym of "tree" |
| Part Holonym | whole element made with the sense | "car" is a part holonym of "window". |
| Entailment | implication of a sense. | "inhale" is an entailment of "smoke" |
| Attribute | quantifier of a sense | "weight" is an attribute of "heavy" |
| Cause | non causative counterpart of a verb | "burn" is a cause of "ignite |
| See Also | alternate/equivalent version of a sense | "many" is a see also of "more" |
| Direct Hypernym | category of a sense | "tree" is a direct hypernym of "palm" |
| Direct Hyponym | type/kind of a sense | "tiger" is a direct hyponym of "cat" |

Table 2: Some of the covered relations in JWNLSIMPLE from getters in JWNL returning a JWNL list.

### 3.5. Usage

Although strictly text based procedures are supplied, JWNLSIMPLE is best used along with the JWNL namespace, which provides proper objects for abstraction of WordNet basic concepts/datatypes, namely for synsets and words.

```
JWNLSimple dbjwnl = new JWNLSimple("db.xml");
dbjwnl.init();
for (Synset ss : dbjwnl.getAllSynsets("word")) {
  // print synonyms of a "word" sense
  for (Word w1 : ss.getWords()) {
    System.out.println(w1.getLemma());
  }

  // print direct hypernyms of a "word" sense
  for(Synset ss1: dbjwnl.getDirectHypernyms(ss)){
    for (Word w1 : ss1.getWords()) {
      System.out.println(w1.getLemma());
    }
  }
}
dbjwnl.close();
```

Figure 1: JWNLSIMPLE usage example.

The example shown in Figure 1 illustrates usage of JWNL datatypes and their property access methods, while searching for synonyms and direct hypernyms of all senses of the word "word" on a database allocated WordNet specified in the "db.xml" argument. The illustrated object life cycle is particularly meaningful when using databases since the connection reuse patch lacks monitoring capabilities on closed or inactive connections (connection pooling).

## 4. Conclusions and future work

Some of the getters in JWNL use WordNet terminology, but return specific data types, therefore requiring traversal/usage knowledge from an NLP oriented developer intending to use WordNet. JWNLSIMPLE reduces this knowledge requirements to basic/essential structures, containing only WordNet related information easily accessible/understood with their API's terminology. These containers are then grouped in Java's standard datatypes with well known efficiency, organization and usage.
JWNLSIMPLE allows a WordNet usage with minimum knowledge of the inner elements/objects that represent the JWNL and/or WordNet model, while other frameworks, like JAWS[14] or JWI[15], force the developer to learn implementation specific objects and their usage, usually through

[14]http://lyle.smu.edu/ tspell/jaws/index.html
[15]http://projects.csail.mit.edu/jwi/

| Relation | Description | Example |
|---|---|---|
| Similar To | characterization of a sense | "calm" is a similar of "quiet" |
| Antonym | opposite of a sense (direct or through a similar) | "naive" is an antonym of "sophisticated" |
| Inherited Hypernym | the category of a sense eventually with other hypernyms in between | "time period" is an inherited hypernym of "May" |
| Full Hyponym | the type/kind of a sense eventually with other hyponyms in between | "magenta" is a full hyponym of "color" |

Table 3: Covered relations in JWNLSIMPLE from getters in JWNL returning a JWNL tree.

examples accompanying the API, even for simple tasks (according to the WordNet model) like synonym retrieval. These frameworks provide distinct features from JWNL which may better suit certain usage scenarios, as such the kind of abstraction provided by JWNLSIMPLE would increase developer's adoption to WordNet and, consequently, to NLP applications.

Mapping an SQL lexicon for JWNL compliance is straightforward with the developed script since all required information slots are defined and can be filled with as much information as available on the lexicon (eventually concatenated for WordNet compliance) allowing access to any SQL lexicon in a WordNet manner.

As future work, we will fully use the functionality provided by JWNLSIMPLE in the task of Question-Answering (QA), as we have already integrated this tool in our laboratory's QA system. This includes, for instance, using the lexical and semantic relations to flexibilize the unification of patterns with sentences in the pattern-based strategy to answer extraction, or as providers of features to feed the machine learning-based question classifier.

## 5. Acknowledgments

This work was supported by Fundação para a Ciência e a Tecnologia (INESC-ID multiannual funding) through the PIDDAC Program funds, and also through the project FALACOMIGO (ProjectoVII em co-promoção, QREN n 13449) that supports Sérgio Curto's fellowship. Pedro Fialho is supported by project CMU-LTI-2527 (from Carnegie Mellon University's Information and Communication Technologies Institute in Portugal) and Ana Cristina Mendes is supported by a PhD fellowship from FCT (SFRH/BD/43487/2008).

## 6. References

Luisa Bentivogli, Andrea Bocco, and Emanuele Pianta. 2004a. Archiwordnet: Integrating wordnet with domain-specific knowledge.

Luisa Bentivogli, Pamela Forner, Bernardo Magnini, and Emanuele Pianta. 2004b. Revising the wordnet domains hierarchy: semantics, coverage and balancing. In *Proc. Workshop on Multilingual Linguistic Resources*, MLR '04, pages 101–108, Stroudsburg, PA, USA. Association for Computational Linguistics.

Davide Buscaldi and Paolo Rosso. 2008. Geo-wordnet: Automatic georeferencing of wordnet. In Bente Maegaard Joseph Mariani Jan Odjik Stelios Piperidis Daniel Tapias Nicoletta Calzolari (Conference Chair), Khalid Choukri, editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, Valentin Tablan, Niraj Aswani, Ian Roberts, Genevieve Gorrell, Adam Funk, Angus Roberts, Danica Damljanovic, Thomas Heitz, Mark A. Greenwood, Horacio Saggion, Johann Petrak, Yaoyong Li, and Wim Peters. 2011. *Text Processing with GATE (Version 6)*. GATE.

Rui Pedro dos Santos Correia. 2010. Automatic question generation for reap.pt tutoring system, July.

C. Fellbaum. 1998. *WordNet: an electronic lexical database*. Language, speech, and communication. MIT Press.

J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarran. 1998. Indexing with WordNet synsets can improve text retrieval. *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*.

Xiaofen He. 2006. A protocol for constructing a domain-specific wordnet ontology for use in lexical-chaining analysis of biomedical texts.

Andreas Hotho, Steffen Staab, and Gerd Stumme. 2003. Wordnet improves text document clustering. In *In Proc. of the SIGIR 2003 Semantic Web Workshop*, pages 541–544.

Armando Stellato Maria Teresa Pazienza and Alexandra Tudorache. 2008a. A bottom-up comparative study of eurowordnet and wordnet 3.0 lexical and semantic relations. In Bente Maegaard Joseph Mariani Jan Odjik Stelios Piperidis Daniel Tapias Nicoletta Calzolari (Conference Chair), Khalid Choukri, editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

Armando Stellato Maria Teresa Pazienza and Alexandra Tudorache. 2008b. Jmwnl: an extensible multilingual library for accessing wordnets in different languages. In Bente Maegaard Joseph Mariani Jan Odjik Stelios Piperidis Daniel Tapias Nicoletta Calzolari (Conference Chair), Khalid Choukri, editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may. European Lan-

guage Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

Rada Mihalcea and Dan Moldovan. 2001. Ez.wordnet: principles for automatic generation of a coarse grained wordnet. In *In Proceedings of Flairs 2001*, pages 454–459.

George A. Miller and Florentina Hristea. 2006. Wordnet nouns: Classes and instances. *Comput. Linguist.*, 32:1–3, March.

George A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41.

Li Shiuan Peh and Hwee Tou Ng. 1997. Domain-specific semantic class disambiguation using wordnet. In *In ACL, Workshop on Very Large Corpora*, pages 56–65.

Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. 2002. Multiwordnet: developing an aligned multilingual database. In *Proceedings of the First International Conference on Global WordNet*, January.

Gangemi van Assem and Schreiber. 2006. Rdf/owl representation of wordnet, w3c working draft.

Piek Vossen. 1997. Eurowordnet - a multilingual database for information retrieval. In *InProceedings of the DELOS workshop on Cross-language Information Retrieval*, pages 5–7.