JMWNL: An extensible multilingual library for accessing wordnets in different languages

Maria Teresa Pazienza[a], Armando Stellato[a], Alexandra Tudorache[ab]

a) AI Research Group, Dept. of Computer Science, Systems and Production
University of Rome, Tor Vergata
Via del Politecnico 1, 00133 Rome, Italy
{pazienza,stellato,tudorache}@info.uniroma2.it

b) Dept of Cybernetics, Statistics and Economic Informatics, Academy of Economic Studies Bucharest
Calea Dorobanţilor 15-17, 010552,
Bucharest, Romania
alexandra.tudorache@gmail.com

Abstract

In this paper we present JMWNL, a multilingual extension of the JWNL java library, which was originally developed for accessing Princeton WordNet dictionaries. JMWNL broadens the range of JWNL's accessible resources by covering also dictionaries produced inside the EuroWordNet project. Specific resources, such as language-dependent algorithmic stemmers, have been adopted to cover the diversities in the morphological nature of words in the addressed idioms. New semantic and lexical relations have been included to maximize compatibility with new versions of the original Princeton WordNet and to include the whole range of relations from EuroWordNet. Relations from Princeton WordNet on one side and EuroWordNet on the other one have in some cases been mapped to provide a uniform reference for coherent cross-linguistic use of the library.

1. Introduction

The success of the original *net of words* which has been originally designed and developed at the Princeton University under the direction of prof. Miller, basing on psycholinguistic theories of human lexical memory, has probably gone beyond its original expectations. Though not being originally realized for computational uses, WordNet has become a valuable resource in the Human-Language Technologies and Artificial Intelligence research areas: WordNet's vast coverage of English words and their organization around the lexico-semantic model which made it a unique resource in its genre, provide the required language structures on which open-domain language processing is based.

Today, *WordNet* is not only the proper name of the original Princeton lexical database, but *wordnet* is used as a common name for denoting lexical resources designed according to the same principles which guided its realization: at present date, the home site of the Global Wordnet Association (http://www.globalwordnet.org/) reports the existence of roughly 50 wordnets for several idioms of the world.

The wide success of WordNet and its enthusiastic adoption in computational linguistics has generated lot of tools and libraries for accessing its content and/or providing extended functionalities for deriving new information from available data (e.g. least common ancestor between sets of synsets, shortest path between synsets through existing lexico/semantic relations etc…). Given the standardization "de facto" which its core model has assumed with respect to its several existing counterparts for different languages, and the importance that multilinguism is achieving in the Information Society, it is surely important to start thinking in terms of *wordnets* related resources (access libraries, applications, visualization tools etc..) more than *WordNet* ones, identify those features which characterize its main model and which are surely replicated along different wordnets, those aspects which – still finding their place in the general model – can change from language to language (or, simply, from version to version), and those which require dedicated effort for being integrated.

Following this objective, we have realized *JMWNL*: An extensible multilingual library for accessing wordnets in different languages. Our goal was to extend John Didion's Java WorldNet Library[i] (JWNL), to support access to the content of the diverse wordnets which have and are arising in these years, accounting for the necessary exigencies that a truly multilingual tool requires.

From the mere point of view of resources compatibility, JMWNL extension focuses on EuroWordNet (Vossen, 1998) lexical database integration, though it can be easily adapted to load all EuroWordNet style resources like BalkanNet (Stamou, et al., 2002) and the recent ItalWordNet (Roventini, et al., 2002).

Furthermore, JMWNL is completely back-compatible with the JWNL library, offering the same API for accessing both EuroWordNet dictionaries as well as the classical Princeton WordNet databases (Miller, Beckwith, Fellbaum, Gross, & Miller, 1993) with no need of updating applications which were already based on JWNL. Princeton WordNet compatibility was also updated to incorporate the new lexical and semantic relations present in WordNet 2.1/3.0: see (WordNet 3.0 official statistics) for an index of changes in the last wordnet versions. The last version of JWNL (1.3) in fact included only pointers and functions updated to WordNet 1.7 while the parallel evolutions that guided the development of the original WordNet up to its version 3.0 on the one side, and EuroWordNet from WordNet 1.6 on the other one, led to a multitude of lexical and semantic relations which needed to be included and, in more than a few cases, mapped in a consistent framework. So, an important task of the project was also the exploration and the mapping of lexical and semantic relations between those defined in WordNet and those in EuroWordNet.

[i] http://sourceforge.net/projects/jwordnet

This document describes the architecture of the JMWNL library, walks through its most important features, discusses some of our design decisions and illustrates how it can be used and customized to build real-world linguistic applications.

2. Requirements and Design Goals

Lexical resources are constantly under development. In this perspective it was not only necessary to extend the present JWNL to encapsulate other lexical resources like EuroWordNet and update it to be compliant with the latest English WordNet changes, but also to give it a more general and extensible architecture.

Objective of JMWNL is to simplify and accelerate common linguistic tasks and at the same time offer support to language-oriented applications for accessing wordnet-like resources for different languages inside an homogeneous and well-structured framework.

One important constraint in the design of JMWNL was to preserve full compatibility with JWNL standard library. So, there is no need to adapt present applications based on the previous version of the library. Another requirement was to load any lexical resource that has the same format as EuroWordNet with independence from language, specific lexical relations and file position/naming, because different version of EuroWordNet for the various languages have slightly different organization schema: some keep all data in one file, other are divided according to part-of-speech, other have a few top synsets which break the synset-word-indexes consistency of the standard WordNet etc.. Last, but surely not the least, dealing with several languages requires dedicated processing steps which cannot in every way be generalized under every aspect. In particular, a library for accessing linguistic resources for different languages should comprehend dedicated morphological analyzers and maintain indexes of lexical objects appropriately, to ensure their correct retrieval.

We extended JWNL keeping in mind these main requirements, thus the only changes (from the user point of view) are in (already existing since the original JWNL) properties and resource files, that are anyway language dependent. This way JMWNL can directly replace the standard JWNL as support for wordnet dependent

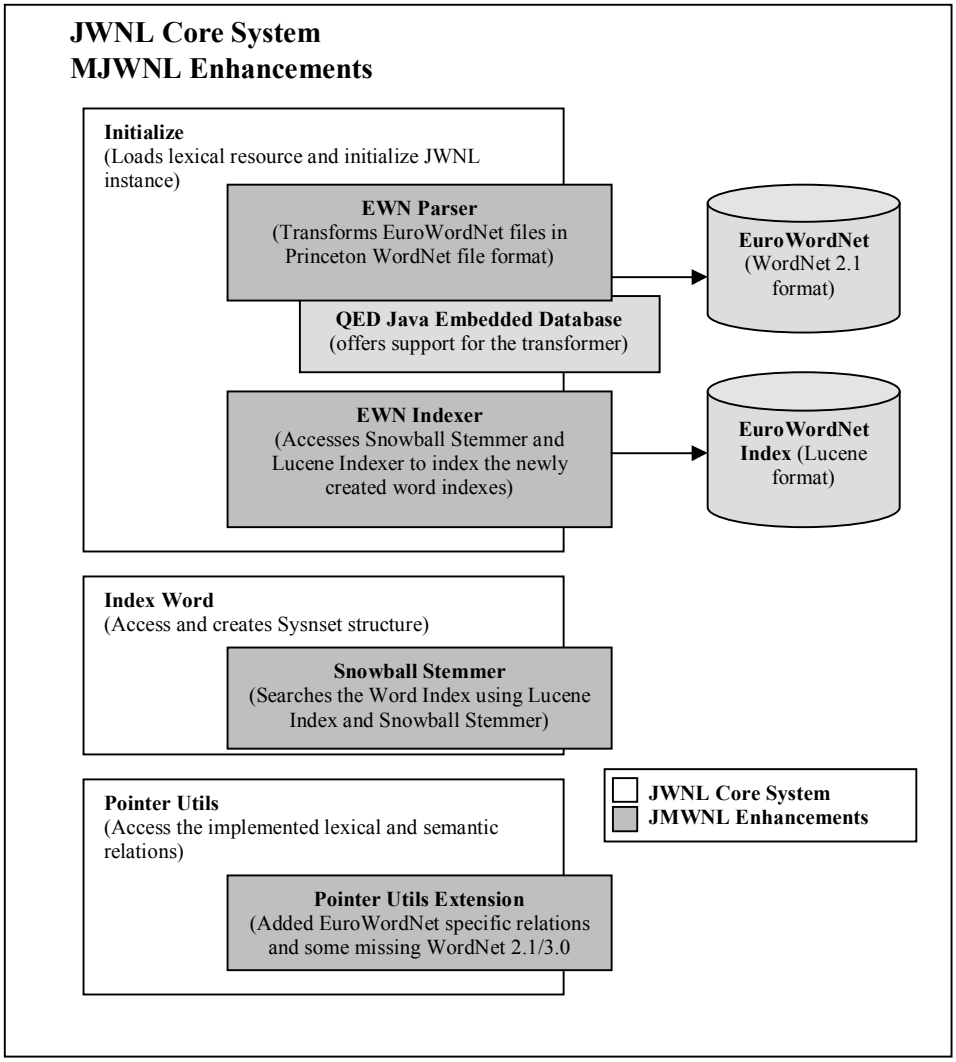

Figure 1: JMWNL Metamodel: the five main components we developed/integrated to enhance JWNL

EuroWordNet Format

```
0 @2511@ WORD_MEANING
  1 PART_OF_SPEECH "n"
  1 VARIANTS
    2 LITERAL "gatto"
      3 SENSE 1
      3 STATUS new
      3 EXTERNAL_INFO
        4 SOURCE_ID 1
          5 TEXT_KEY "0-1"
    2 LITERAL "micio"
      3 SENSE 1
  1 INTERNAL_LINKS
    2 RELATION "has_hyperonym"
      3 TARGET_CONCEPT
        4 PART_OF_SPEECH "n"
        4 LITERAL "felino"
          5 SENSE 1
    2 RELATION "has_hyponym"
      3 TARGET_CONCEPT
```

```
4 PART_OF_SPEECH "n"
        4 LITERAL "gattomammone"
          5 SENSE 1
.....
    2 RELATION "role_agent"
      3 TARGET_CONCEPT
        4 PART_OF_SPEECH "v"
        4 LITERAL "miagolare"
          5 SENSE 1
  1 EQ_LINKS
    2 EQ_RELATION "eq_synonym"
      3 TARGET_ILI
        4 PART_OF_SPEECH "n"
        4 WORDNET_OFFSET 1457819
    2 EQ_RELATION "eq_near_synonym"
      3 TARGET_ILI
        4 PART_OF_SPEECH "n"
        4 WORDNET_OFFSET 1457160
```

WordNet Format

```
01363515 00 n 02 gatto 1 micio 1 012 ra 00350450 v 0000 ~ 01922686 n 0000 ~ 01915820 n
0000 ~ 01915767 n 0000 ~ 01908172 n 0000 ~ 01908121 n 0000 ~ 01908069 n 0000 ~ 01908002 n
0000 ~ 01907950 n 0000 ~ 01327668 n 0000 ~ 01327612 n 0000 @ 01360193 n 0000
```

Figure 2: Transformation of EuroWordNet data into WordNet 3.0 format

applications requiring changes only on configuration data, but maintaining and exposing the same interface (i.e no hard-coding changes).

3. Architecture of JMWNL

JMWNL can be used to explore lexical databases organized following both the original Princeton WordNet and the EuroWordNet format (e.g. EuroWordNet, BalkanNet and the recent ItalWordNet) but also be easily extended to read new file formats and syntaxes representing information organized after the same general wordnet model.
The library includes a collection of classes and functions to access lexical resources and to explore lexical and semantic relations. It is possible to access and explore direct semantic and lexical relations (hypernymy, hyponymy, antonymy, cause etc…) or even adopt and/or operate on recursive patterns for their exploration.

3.1. JMWNL Extensions

As illustrated in Figure 1, the JMWNL library extends JWNL with components and modules for transforming and accessing data, appropriately stemming words from the different databases according to their language and indexing these stems for efficient retrieval.
To extend the capabilities of JWNL to work with EuroWordNet data we integrated five main components:

1. EuroWordNet file transformer, which translates EuroWordNet files into their standard WordNet counterparts
2. Snowball stemmer (http://snowball.tartarus.org/index.php), an open source library providing implementations for diverse stemming algorithms (Farber, Griswold, & Polonsky, 1964; Lovins, 1968) for several languages
3. Lucene (Hatcher & Gospodnetic, 2004): an indexing engine for automatically indexing the stemmed words of the loaded wordnets
4. QED JAVA Embedded Database (http://www.quadcap.com/products/qed/docs/index.html) as a support for source syntax transformation.
5. Extension to the main component for new access functions to EuroWordNet and WordNet 3.0 Lexical and semantic Relations.

3.2. Main tasks of JMWNL development

For this project we identified four key development tasks: JWNL enhancement to support all WordNet 2.1 /3.0 lexical and semantic relations, the validation of the implementation of lexical relations in JWNL, development of JWNL extension for EuroWordNet and multi-instance support for loading multiple lexical resources (this last one is still under development, since it requires deep changes in the nature of JWNL).
The first two tasks and the fourth one involved development of an updated version of the original library, while the third one required new add-ons (the *transform* interface and its first implementation for EuroWordNet file formats) and the integration of other external components.

3.3. Extension to support WordNet 2.1/3.0

To fully exploit the improved content of the new WordNet versions from 2.0 to 3.0, we added support for relations like: instance hypernymy (represented by the "@i" pointer character), instance hyponymy ("~i" pointer

```
Causes of "ottenere":
  [PointerTargetNode: [Synset: [Offset: 580947] [POS: noun] Words: acquisizione] null]

Direct near synonyms of "bello":
[PointerType: [Label: near synonym ] [Key: ns] Applies To: noun, verb, adjective, adverb]
  [PointerTargetNode: [Synset: [Offset: 72929] [POS: adjective] Words: avvenente, attraente
-- (detto di ciò che attrae)] null]
  [PointerTargetNode: [Synset: [Offset: 39497] [POS: adjective] Words: pregevole,
apprezzabile, stimato, lodevole -- (Che è di qualità, pregevole; Un lavoro pregevole)]
null]

 Direct Hyponyms of "albero":
  [PointerTargetNode: [Synset: [Offset: 1509993] [POS: noun] Words: cacao] null]
  ……
  [PointerTargetNode: [Synset: [Offset: 1551817] [POS: noun] Words: sughero, sughera] null]
```

Figure 3: Examples of JMWNL output on the Italian EuroWordNet lexical resource

character), derivationally related form ("+") and troponyms ("~"). We also needed to modify access methods for other past relationships and adapt them according to needed substitutions (as for *troponymy*, which replaced *hypernymy* for verbs) and mappings between WordNet and EuroWordNet relations.

3.4. Extension for EuroWordNet

Since EuroWordNet file format is strongly different from the original WordNet one, we had two options: to modify the library in order to accept EuroWordNet (EWN from now on) file format or to convert EuroWordNet files into WordNet format.

The main difference between the two formats lies in the way information is stored. At a logical level the information is equivalent, as it is compliant with the wordnet lexical database structure (in short: lexical information organized around sets of synonymical words: synsets, which are in turn connected through semantic relationships, plus the definition of specific lexical relationships between couples of synset-word pairs). At the physical level, EuroWordNet interchange textual files are centered upon words definitions offering an high human readability, but are less prone to be used natively for efficient retrieval (EuroWordNet is distributed with a specific DB and an associated application for being searched) than their original WordNet counterpart, which adopt symbols and pointers to the physical offsets (in bytes) of word definitions to describe relations. This allows a better indexing and a faster access to data.

Given this premises we have chosen to define a generic *transformation* interface (which could be arbitrarily implemented for addressing other WorldNet-like resources) and to realize a first implementation for converting EWN files to WordNet 3.0 format.

We also integrated the Lucene indexing engine and Snowball Stemmer to index and stem the EuroWordNet lexical resource. This way JMWNL is capable of loading and working with EuroWordNet format data for the following languages: English – with different stemming implementations available, like Porter's (Porter, 1980) or Lovin's (Lovins, 1968) stemmers – French, Spanish, Portuguese, Italian, Romanian, German, Dutch, Swedish, Norwegian, Danish, Russian, Finnish, Hungarian, Turkish. Future implementations will include a generic interface for lemmatization, to obtain precise retrieval of proper lemma from declined/inflected forms, through lemmatizers for all the EuroWordNet/BalkaNet languages are not easily available in the opensource community.

4. Development Issues

One major issue was the mapping of lexical relations and semantic relations found in EuroWordNet and WordNet.

Due to the intrinsic nature of lexical resource development and of the work over the years of multiple research groups from different countries, there are some major differences between the relations defined in EWN and WordNet.

EuroWordNet is richer in relations than WordNet and, though there is a lot of overlap between them, a one-to-one matching is not possible.

In this perspective we mapped the majority of relations between the two lexical databases and added all the remaining relations found in EuroWordNet so that no information were lost.

For WordNet compatibility we also had to collapse the private nouns and the nouns files into the single NOUN POS (though this helped in distinguishing simple hypernymy from instance hypernymy between EWN synsets ported to the standard wordnet format). This operation is done when the files are first parsed.

Like for any information system another problem is represented by the limitedness of the processing power versus the great amount of data to be processed. In particular, the transformer requires massive processing power and memory because it has to calculate and then replace all the new offsets to match WordNet 2.1/3.0 format and then properly attach all the relationships between the newly generated offset-identified synsets.

For example the Italian WordNet resource has nearly 45.000 synsets distributed along the 4 PoS, as in Table 1.

Due to the great amount of data (and related memory

POS	No. of Synsets
Nouns	33605
Verbs	8816
Adjectives	2246
Adverbs	199
TOTAL	**44866**

Table 1:Italian EuroWordNet synset distribution

requirements) to process during the transformation, we opted for the integration of an embedded database to support the generation of the new WordNet format files, thus allowing for a robust and not memory-demanding solution while keeping the simplicity of J(M)WNL, which can be still considered a library and not a complex component, requiring no particular installation procedure.
Figure 2 shows an example of transformation from EuroWordNet data to WordNet Format for the synset containing "cat" (gatto, micio)
To increase speed we also provided the option to handle all the conversion entirely in memory. This requires 1 GB of RAM reserved to the Java process, quite demanding now but probably an acceptable choice for immediate future technologies as it is much fastest than the DB one.
The transformation is done only once when a given EuroWordNet resource is loaded for the first time. Our tests revealed that on an average notebook with 1 GB of memory and 2 GHz of processor speed, the process for the biggest dictionary (the Italian one) takes around one hour and fifteen minutes (when using the DB). Once produced, the transformed resource and the wordstems index can also be ported on different machines, thus avoiding this process even for the first run of the library on a newly installed machine. We could not directly provide the processed data on the web since the original EWN databases can be obtained under license from the European Language Resources Association, ELRA (http://www.elra.info/), thus users in possess of a licensed version, can launch the transformer once for each language wordnet they own.

5. Technical details

5.1. Configuration and use

Since JMWNL was developed as a library to be included in linguistic software the configuration is done relatively easy with the help of two kinds of files: Properties File and Resource files. The configuration of these files depends on the language of the resource to be loaded and on the compliance of its source files (be it original WordNet files or converted EWN ones) towards one of the versions of WordNet.
Every language/wordnet has its own properties and resource file, with the first one defining configuration options for the library (the language, path of resources, database support, stemmer to be used etc…) and the second one providing template static entries like the symbols of the pointers, the names of the relations, for each language. Also you will found the definition of exceptions and other parameters like verb frames. Exception messages regarding Synsets creation and usage are defined in the main PrincetonResource file, which is common to all the loaded resources.

5.2. Availability

JMWNL library is currently available for download at our project site: http://ai-nlp.info.uniroma2.it/software/jmwnl and it will possibly move to the original JWNL site on sourceforge, as the next version of the original library.

6. Conclusions and Future Work

In this paper we have presented our effort in extending the most popular software library for WordNet with multilingual capabilities and support for different wordnet versions.
JMWNL can be used as library if integrated in a linguistic tool or can be tested stand alone using the java interpreter and the test/query files provided. We plan to develop JMWNL to include multi-instance support for loading in parallel multiple wordnet resources. This way, linguists, developers and researchers will have the opportunity to explore and work on multilingual lexical resources and also to develop multilingual applications requiring concurrent access to wordnets in different languages, or just explore multiple versions of the same wordnet to analyze differences along its development. Exploitation of the EuroWordNet Interlingua Index (ILI) has been deliberately ignored in this version, since it will not necessarily be adopted in other future wordnets, but we plan to evaluate its worthiness and relevance for future versions of the library, by providing cross-lingual navigation of different wordnets through anchors in the ILI.

7. References

Farber, D., Griswold, R., & Polonsky, I. (1964). SNOBOL, a string manipulation language. *Journal of the Association for Computing Machinery , 11*, 21-30.

Hatcher, E., & Gospodnetic, O. (2004). *Lucene in Action.* Manning.

Lovins, J. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics , 11*, 22-31.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1993). *Introduction to WordNet: An On-line Lexical Database.*

Porter, M. (1980). An algorithm for suffix stripping. *Program , 14*, 130-137.

Roventini, A., Alonge, A., Bertagna, F., Calzolari, N., Marinelli, R., Magnini, B., et al. (2002). ItalWordNet: A Large Semantic Database for the Automatic Treatment of the Italian Language. *First Internationa WordNet Conference.* Mysore, India.

Stamou, S., Oflazer, K., Pala, K., Christoudoulakis, D., Cristea, D., Tufiş, D., et al. (2002). BALKANET: A Multilingual Semantic Network for the Balkan Languages. *International Wordnet Conference*, (p. 12-14). Mysore, India.

Vossen, P. (1998). *EuroWordNet: A Multilingual Database with Lexical Semantic Networks.* Dordrecht: Kluwer Academic Publishers.

Vossen, P. T., & Fellbaum, C. (n.d.). *Wordnets in the world*. Retrieved March 26, 2008, from Global WordNet Association: http://www.globalwordnet.org/gwa/wordnet_table.htm

WordNet 3.0 official statistics. (n.d.). Retrieved from WordNet Site: http://wordnet.princeton.edu/man/wnstats.7WN