Data Representations for WordNet: A Case for RDF

Alvaro Graves and **Claudio Gutierrez**
Computer Science Department
Universidad de Chile
Av. Blanco Encalada 2120
Santiago, Chile
agraves@dcc.uchile.cl
cgutierr@dcc.uchile.cl

Abstract

This paper discusses current versions of WordNet from a data modelling perspective. We show that these versions do not consider basic data model desiderata for their design, like flexibility, extensibility and interoperability. We claim that a data model for WordNet must also consider the inherent network structure of WordNet data. Thus we make the case for an RDF model for WordNet and present a concrete version of WordNet in RDF format.

Introduction

In their classic introduction to WordNet, Miller [Miller et al., 1993] state that it is "a proposal for a more effective combination of traditional lexicographic information and modern high speed computation." The original goal of the WordNet project was to improve the "tedious and time-consuming" labor of finding the information in standard alphabetical procedures for organizing lexical information. The underlying argument is that with the advent of computers, we are not anymore bound to the *data structure* that a book or classical printing methods force us. On the contrary, computer technology allows us to build the data structures that best resemble the *conceptual structure* of the problem we are modelling. This allows us humans to browse, navigate and retrieve the information in flexible and unpredictable ways, tasks which were impossible to do with hard copies. Today WordNet is available in a variety of formats, languages and platforms [Miller et al., 2005], having different features and interfaces depending on the objectives for what they were created. WordNet is presented as a software package, which bounds together the data (files in some codification) and the applications.

From a classical data management point of view current versions of WordNet present several drawbacks. Among the most important are the lack of modularity (blurred distinction of operational and data features), the lack of integrity constraints (no type or consistency enforcement), and the nonexistence of the notion of view (no notion that data of different applications and levels of aggregation are simple "views" of a standard data model). Several problems arise. For example, natural questions like checking if a representation is faithful or determining if two versions are the same are almost impossible to answer.

More importantly, one can analyze current WordNet versions from a pure data modelling point of view. A data model is a collection of conceptual tools for describing the real-world entities to be modelled and the relationships among these entities [Silberschatz et al., 1996]. Existing models for WordNet were devised for specific applications. Wordnet is clearly going beyond the objectives the creators had in mind, as occurs with any interesting data source. Dozens of different applications [Miller et al., 2005; Mihalcea,] are using Wordnet. New features are discovered, and last, but not least, updates are being made periodically. More importantly, current trends in information management indicate that one should design data to allow machines to process it without human intervention. This is the core idea of the so called Semantic Web [Berners-Lee et al., 2001]. Thus, a data model for WordNet should be extensible (permit to add new features without modifying essentially the current model), interoperable (independent of hardware platforms, operating systems and software) and flexible (foresee new uses of the data). The essential condition for this to be possible is that the data structure be as close as possible to the conceptual structure being modelled.

Another data model consideration is that Wordnet was created as a semantic network of word meanings (and word forms). This amount to say that the structure of Wordnet at a conceptual level is a directed graph with labeled nodes and arcs. This intrinsic network structure of Wordnet is reassured by the discovery that Wordnet has the main characteristics of a complex network [Sigman and Cecchi, 2001]. Hence, a data model where the data structures are graph-like and which facilitates data manipulations and queries over this graph structure would be a natural choice to model Wordnet. Furthermore, the data model for WordNet must include features for modelling concepts like "Noun", "Adjective" as subclasses of "Word" that is, must consider notions of subclass and inheritance.

A natural question arises: *What is the best data model for Wordnet?* or more modestly, *what is a good data model for Wordnet?* In this paper we discuss this issue by reviewing and analyzing current versions, and laying arguments for a version of Wordnet modelled using the *Resource Description Framework* (RDF) [Consortium, 2004], a proposal of the Web Consortium for modelling semantics of data which meets precisely the above considerations.

Petr Sojka, Key-Sun Choi, Christiane Fellbaum, Piek Vossen (Eds.): GWC 2006, Proceedings, pp. 165–169.

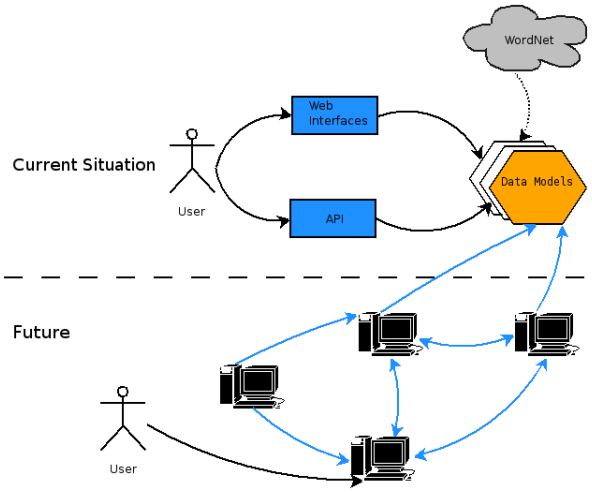

Figure 1: Different ways to access WordNet. Above the line, the current situation where humans process directly the data via interfaces. Below the line, the situation where computers use WordNet data independent of humans.

The paper is organized as follows. In the first section we analyze current representation of Wordnet. In the second section, we introduce RDF and argue why such data model is appropriate for modelling Wordnet. Then in the third section we present the RDF model, and discuss issues that arouse in our implementation. Finally, in section four we summarize and present perspectives and uses of this RDF version.

1 Representations of WordNet

There are several projects related with WordNet [Miller et al., 2005]. They correspond essentially to different forms of accessing and processing Wordnet: Web and human interfaces and application program interfaces (API) (see Figure 1). Roughly they can be grouped as follows.

1.1 Official versions of WordNet

Strictly speaking, there are two official versions of WordNet publicly available.

The data, although in separate files, is bound to an operative system or a particular type of software, and cannot be processed by other applications unless a human interprets the semantics of the files. Currently, the are two core releases, the Windows and the Prolog versions.

In the Windows version is a package including a "WordNet browser, command-line tool, and database files with InstallShield self-extracting installer.". The data component is organized in a set of plain files. Each file is a list of ordered records (one per line) containing several fields. Queries are processed by essentially doing binary search over these records. The main deficiencies of this version are: Data and functionality are not modular; No support for semantics (and typing) of the data; No support for network related queries; Data is not in a standard language or format. In the case of the Prolog version, WordNet is composed of a set of Prolog facts. In this sense, it is more like a knowledge base, with Prolog as the deductive engine. Although in this case there is more modularity, and the data format is that of Prolog (a standard language), the drawbacks are similar to the previous case. A typical example is the semantics given to the binary predicate `vgp`: "the operator specifies verb synsets that are similar in meaning and should be grouped together when displayed in response to a grouped synset search." Note the close binding of the data structure to a particular type of query.

1.2 Web Interfaces

[Miller et al., 2005] These interfaces allow to consult the database of WordNet through the Web. The input as the output are made for humans. As positive aspects we can mention that it relieves the user to install software in his/her computer, and no computer background is necessary to use it. But this version makes impossible automatization of WordNet query and retrieval processes, and cannot be plugged to other software. Additionally, the user cannot navigate (at least in the current versions) the semantic network.

1.3 Application Programming Interfaces (API)

[Miller et al., 2005]. APIs provide a layer between the user and the data. In this sense is not necessary for the developers to be aware of the organization and structure of the raw data to create new applications, as in the Windows version. But they have drawbacks too. First, they are tied to a fix version of WordNet. To update the WordNet version, it is necessary to develop a new version of the application from scratch. Second, APIs are specific, making it difficult to add new features, and so restricting its functionality. Third, these kinds of applications mix the data model with the functionalities they provide, with all the problems this brings to extensibility and interoperability.

1.4 WNconnect

This application deserves a special mention. WNconnect [Fong, 2003] builds networks which relates different words via paths in the network. Although is one of the few applications that incorporate this natural network feature, its main problem is the lack of modularity between data and application. This application uses WordNet 1.7 and is an issue its update to current versions. For the same reason, it is not possible to add new functionalities unless modifying the source code of the whole application.

1.5 Other data models

There are some data models of WordNet based in relational databases (MySQL, PostgreSQL). They take advantage of the general purpose engines made for this technologies. A standard database model seems to be natural choice for WordNet. The point here is if the relational model is the best suited for the needs of WordNet. The answer is no because the relational model performs poorly on network structures, especially for queries involving paths, neighborhoods [Angles and Gutierrez, 2005]. Additionally, although there is support for metadata in the form of the schema, it is not enough to describe a network structure.

1.6 XML

The eXtended Markup Language [Bray et al.,] was created to exchange data between different applications over the Web. In this sense, XML is a primary candidate for modelling the data of WordNet. In fact, there is a XML project over WordNet at the University of Texas at Dallas [Moldovan et al., 2003]. This project is not intended for a complete representation of WordNet. It is focused in the parsing and formalization of the glossaries of WordNet. For example, it has no lexical relations, but synsets and their glossaries parsed. Among its uses is the disambiguation of meanings of words, question answering and information retrieval.

Having a complete and standard XML version of WordNet would be a great advance over current versions, fulfilling several items of the wish-list of a good model. But there are two important aspects that will be necessarily missing: the network structure and the semantics of the data. XML, by its design, performs very well with data which has tree-like structure (e.g. documents, Web pages), but not with data with network structure. Additionally, if one wants to express semantics of data, and model simple inheritance features in an interoperable and extensible way, the obvious choice is RDF or OWL [McGuinness and van Harmelen, 2004].

2 Why a RDF representation

The Resource Description Framework (RDF) [Consortium, 2004] is a recommendation of the W3C, oriented to represent highly interconnected information. An atomic RDF expression is a triple, in the form subject-object-predicate. A general RDF expression is a set of such triples, which can be naturally considered as a labeled graph. According to this, the syntax of RDF reflects a graph data model. Additionally, it has support for describing inheritance of classes and properties. A good introduction is the *Primer* [Manola and Miller, 2004].

The broad goal of RDF is to define a mechanism for describing resources that makes no assumptions about a particular application domain, nor defines (a priori) the semantics of any application domain. RDF is domain neutral and models information with graph-like structure. Examples of its use are, Genome[1], Open Directory[2] and Web data. One of the main advantages (features) of the RDF model is its ability to interconnect resources in an extensible way. Thus, the notion of *connectivity* of resources appear as a central one.

2.1 RDF representation for WordNet

The choosing of RDF is based on several reasons. First, it is a standard for the Web, focused in description of resources (in this case words and its lexical relations). RDF was designed with the aim to support metadata and semantics in a native way, i.e. relations and properties, inheritance like classes and subclasses, and so on. Second, RDF has a natural structure of network, and is ideal to represent data and metadata with that structure. Third, another advantage of RDF is its extensibility; it is easy to add new functionalities or data to this schema. Fourth, the schema describing the structure of this representation can be accessed in the same way as the data, so it does not add complexity for new applications. Finally, the community of developers, designers (among others) for RDF applications is growing each day. This make easy to find support for new developments and maintenance.

2.2 Sufficiency of RDF for WordNet

A question that arises at this stage is the sufficiency of RDF as a format for WordNet. We can summarize the requirements of WordNet as: (a) Relations between entities, described as semantic relations as antonyms, meronyms and so on; (b) Notion of class, as for words as synsets; (c) Notion of hierarchy of classes, like an adjective word as a subclass of word; and (d) Notion of instance and type, meaning that some entity has a type of some kind. All these requirements are accomplished by RDF as a modelling language [Consortium, 2004]. Also, RDF introduces another useful characteristics, like hierarchies among properties (for example, different types of meronyms as subproperties of a general meronym property) and comments.

However RDF does not support other functionalities, like inverse of relations. This means that one relation is the inverse of another. This functionality is desirable and not critical because we always can search for the object in the subject-predicate-object structure of RDF. This would be interesting for hyponym/hypernym and meronym/holonym relations. The more expressive language OWL, *Web Ontology Language* [McGuinness and van Harmelen, 2004], which allows to define ontologies, provides this an other functionalities, for example cardinality, basic set operations, and the possibility to define features like transitivity and functionality of binary relations.

The drawback of using a more expressive language like OWL is the addition of unnecesary (for WordNet) computational complexity and additional difficulty to users and developers to program new applications. In the worst scenario, using all the expresiveness of OWL the search may be undecidable [Antoniou and van Harmeten, 2004].

Due to the fact that WordNet can be expressed completely in RDF, we consider that it is not worth adding the additional complexity to the user and to the automatization of reasoning that OWL brings.

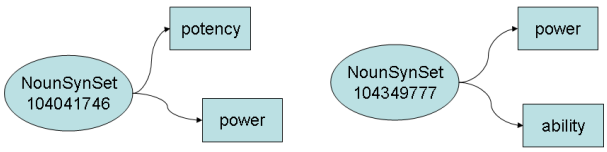

Figure 2: Example of the schema by S. Melnik. The words are considered as labels. Hence it is not possible to relate directly two synsets using the same word. In this case, the word “power” occurs as two different labels with no semantic relation between them.

[1] http://www.affymetrix.com/community/publications/affymetrix/tmsplice/index.affx

[2] http://rdf.dmoz.org/

2.3 Previous RDF representations

In the year 2000, Sergei Melnik made an initial representation in WordNet in RDF [Melnik, 2001]. It consisted in a set of nouns, the glossary and the hyponym and similar-to relations. Also a schema for this representation was provided. However this work is stalled and unfinished. In this schema the synsets are classified as nouns, verbs, adjectives, adverbs and satellite adverbs. All of them are subclasses of the "LexicalConcept" class. The words are defined as "wordForms" as several relations, like meronyms, seeAlso, and others. The only lexical relations defined are antonyms, similarity, hyponyms and a definition of glossary. One drawback of this schema is that it does not take into account the polysemy, i.e., each word is just a label, and is not considered like an entity that can be related with several synsets. For example, there is no way to discover that "power" has several meanings unless all the data is searched (see Figure 2).

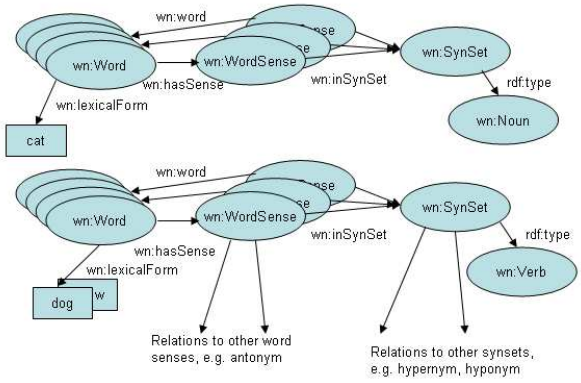

Figure 3: Diagram of the schema by WordNet Task Force.

In the year 2004 the *WordNet Task Force* [Gangemi, 2004b] developed a new schema [Gangemi, 2004a]. This is a better approach than Melnik's to the natural structure of WordNet. An important feature is the addition of the notion of WordSense, i.e., the use of a word in some sense (see Figure 3). For example, the word "power" can be used as 'a physic capacity of work measurable in watts' and also in the sense of 'physical strength'. WordSenses can be thought of as weak entities in databases theory. Another requirement of the WordNet model, is the necessity to create distinct kinds of synsets to differentiate Nouns from Adjectives and so on. The Inheritance mechanism of RDF is used to accomplish this goal, i.e. a generic class "synset" and a set of subclasses, like "NounSynSet", "AdjectiveSynset" are created.

The W3C WordNet project is still in the process of being completed, at the level of schema and data. In fact, the current schema is not usable due to syntax errors, and is incomplete, e.g. relations like "participleOf" are not defined, although there are some discussions and comments in the code. Other relations like "attribute" have their range and domain not yet defined. There is no version of WordNet data using this schema.

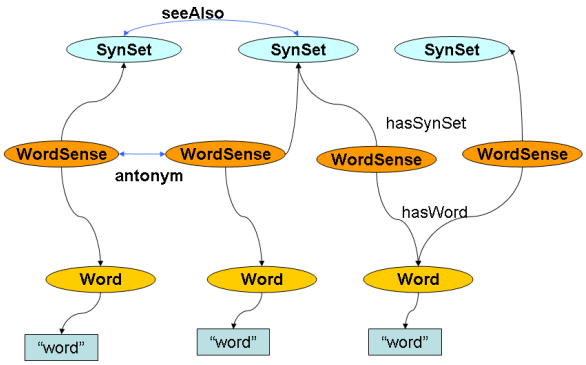

Figure 4: Schema of RDF representation. The model is composed by three layers: Word layer, WordSense layer and SynSet layer. Because of polysemy, a word can be related with various SynSets through several WordSense. Also, it is possible for a SynSet to be related with several Words through WordSenses. Finally, the Word node is related with a label using the same word.

3 The RDF Representation

3.1 Modelling WordNet

We based our work on the model developed by the WordNet Task Force [Gangemi, 2004b]. They created an initial schema [Gangemi, 2004a], but this project has been stalled for over one year since then. We completed and slightly modify some features of this model. The most noticeable difference is that we modelled the Word node identifier as the same word for performance reasons, and leaving as well the label which indicates the same the word. This avoids the necessity of definition of new identifiers for each word. For example, in Figure 3, the label cat was inserted directly in its corresponding node *wn:Word*.

We based our work in the Prolog version of WordNet 2.0. The main problem was to define the schema of WordNet. In this version, there are three layers, as in the W3C schema. The first layer is composed of a set of nodes which are subclasses of class "Word". It is important to note that the words are represented by nodes in the graph and are not just labels. This allow to represent correctly the polysemy inherent in WordNet. The WordSenses layer is the link between a Word and a SynSet. The SynSet layer is composed by a set of "NounSynSet", "AdjectiveSynSet", "AdjectiveSatelliteSynSet", "VerbSynSet" and "AdverbSynset", which are subclasses of SynSet. The lexical relations are located in the second and third layer. Examples of this are antonyms and seeAlso relations. See Figure 5. Moreover, each SynSet has several subclasses, like NounSynSet or AdjectiveSynset. The above is useful to define specific domains and ranges for every relation present in WordNet.

Another problem to be addressed, was the cleaning, parsing and validation of the data. Initially, we developed a set of parsers in Python using 4Suite library for the RDF management. However the prohibitive time required to parse one entire file made it non feasible. After several attempts using

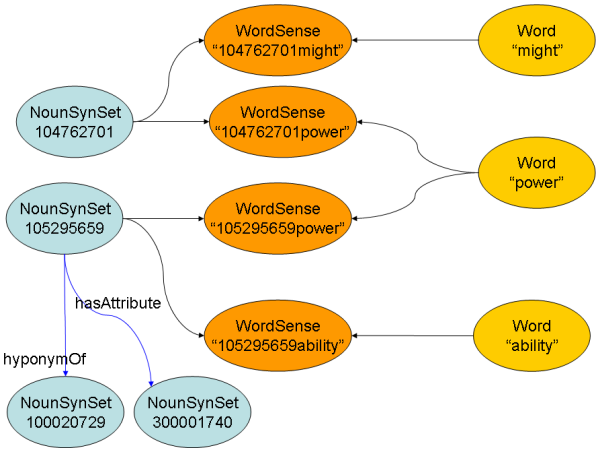

Figure 5: Example of our RDF representation.

other languages we decided to create the RDF/XML structure from scratch using Perl. This decision improved the time required to obtain the results, making easy the debugging.

Finally, we compared our new version with the Prolog version, to ensure its completeness. This was made moving our version to a relational database (MySQL), and checking that no entities were missed.

3.2 The RDF version of WordNet

We left the representation in separated files for each relation. Also, there is a .bz2 compressed bundle. This files are available at `http://www.dcc.uchile.cl/~agraves/wordnet`

Also, there is a version of the RDF tuples of this representation in Berkeley Database format available in `http://alumnos.cadcc.cl/~agraves/wordnetDB.tar.bz2`

4 Conclusions

The main advantage of RDF for representing WordNet is allowing to represent it as a network in a natural, simple and lightweight way. Also, this raise new possibilities for visualization and for asking new kinds of queries. Another advantage is the accessibility through the web, allowing different applications to consult the data. Even more, WordNet is expressed now in a standard way for the semantic web; this will permit the use of semiautomatic agents for more complex searches in the future. Also, the growing community around RDF language opens new possibilities of collaboration in the developing of new applications and support for WordNet.

Among the possible applications are: Establishing long-term relations between words or synsets; Search for neighborhood and semantic chains; Availability of WordNet data for semiautomated agents; Implementation of Web Services.

Acknowledgements

This paper has been supported by Proyecto FONDECYT No. 1030810. Claudio Gutierrez also acknowledges Proyecto Nucleo Milenio, Center for Web Research, P04-067-F.

References

R. Angles and C. Gutierrez. 2005. Querying rdf data from a graph database perspective. *2nd. European Semantic Web Conference (ESWC2005), Heraklion, Greece*, 3532:346–360, May.

Grigoris Antoniou and Frank van Harmeten. 2004. *A Semantic Web Primer*. MIT Press.

Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The Semantic Web. *Scientific American*, May.

Tim Bray, Jean Paoli, and C. M. Sperberg-McQueen. Extensible Markup Language (XML) 1.0, W3C Recommendation 10 February 1998. `http://www.w3.org/TR/1998/REC-xml-19980210`.

Web Consortium. 2004. Resource description framework (rdf), `http://www.w3.org/rdf/`. October.

Sandiway Fong. 2003. Wnconnect software, `http://dingo.sbs.arizona.edu/~sandiway/wnconnect/`, November.

Aldo Gangemi. 2004a. WordNet in RDFs and OWL, `http://www.w3.org/2001/sw/bestpractices/wnet/wordnet-sw-20040713.html`.

Aldo Gangemi. 2004b. WordNet task force, `http://www.w3.org/2001/sw/bestpractices/wnet/tf.html`, July.

Frank Manola and Eric Miller. 2004. Rdf primer, `http://www.w3.org/tr/rdf-primer/`. February.

Deborah L. McGuinness and Frank van Harmelen. 2004. Owl web ontology language overview, `http://www.w3.org/tr/owl-features/`.

Sergey Melnik. 2001. RDF representation of WordNet, `http://www.semanticweb.org/library/`, February.

Rada Mihalcea. WordNet bibliography, `http://engr.smu.edu/~rada/wnb/`.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1993. Introduction to wordnet: An on-line lexical database. August.

George A. Miller, Christiane Fellbaum, Randee Tengi, Susanne Wolff, Pamela Wakefield, Helen Langone, and Benjamin Haskell. 2005. WordNet related projects, `http://wordnet.princeton.edu/links`.

Dan I. Moldovan, Orest Bolohan, and Adrian Novischi. 2003. Extended WordNet, `http://xwn.hlt.utdallas.edu/index.html`.

Mariano Sigman and Guillermo A. Cecchi. 2001. Global organization of the WordNet lexicon. November.

Avi Silberschatz, Henry F. Korth, and S. Sudarshan. 1996. Data models. *ACM Computing Surveys*, 28(1):105–108.