WordVenture – COOPERATIVE WordNet EDITOR
Architecture for Lexical Semantic Acquisition

Julian Szymański
Gdańsk University of Technology, Narutowicza 11/12, 80-952 Gdańsk, Poland

Keywords: Linguistic semantic networks, WordNet, Collaborative editing, Lexical semantics, Wikipedia.

Abstract: This article presents architecture for acquiring lexical semantics in a collaborative approach paradigm. The system enables functionality for editing semantic networks in a wikipedia-like style. The core of the system is a user-friendly interface based on interactive graph navigation. It has been used for semantic network presentation, and brings simultaneously modification functionality.

1 INTRODUCTION

WordNet (Miller et al., 1990b) is one of the largest semantic lexicons of English. It has been developed since 1985 by the Cognitive Science Laboratory at Princeton University. Its authors, based on theories of human cognition, try to reflect all linguistic dependencies between concepts in a common lexical database. The WordNet team has been working on a semantic dictionary for over 22 years. Nowadays[1], the dictionary contains about 155287 words, organized in 117659 synsets (meaning representations), and includes 206941 pair words – meaning. Introduction of all words with their connections, as well as examples of their usage in language, requires a lot of human work, however the WordNet team has only seven members. The WordNet project has been supported by plenty of grants, which brought together 3 millions dollars. Currently the third release of the WordNet lexical database is available at the project website[2]. WordNet develops as a research project in a closed academic environment. The first version of the dictionary appeared in 1993, and now a third version is available. The dictionary is publicly available, but its modification is restricted to internauts. Probably, the reason for that, is the fact that the lexicon is organized as a set of text files in a specific format, which makes it hard to apply cooperative approach for WordNet development. Lack of cooperative editing functionality is the biggest barrier to scale-up semantic database.

The most well known application of a cooperative approach for gathering data is Wikipedia. The project has experienced great interest from the Internet community which brought many positive results. Wikipedia has been developed since 2001 by volunteers from all over the world. Currently, the Wikipedia initiative is supported by almost 75000 people, working on over nine million articles written in 125 languages. The largest set of articles is available in English, and contains over 2 million articles.

Nowadays, a lot of projects has been created on the basis of WordNet[3]. They use semantic dictionary as a core knowledge base about language, what enables to implement elementary linguistic competences in a machines.

Some of the implementations do the mapping from WordNet files to other models, especially relational. This can be used to enable a cooperative editing approach.

2 DESCRIPTION OF WordVenture SYSTEM

A WordVenture portal[4] has been developed at the Gdansk University of Technology at the Faculty of Electronics, Telecommunications and Informatics. It provides mechanisms for simultaneous work on lexical dictionaries for distributed groups of people and enables cooperative work on aWordNet lex-

[1] ver. 3.0
[2] http://wordnet.princeton.edu
[3] see: related projects http://wordnet.princeton.edu/links
[4] http://wordventure.eti.pg.gda.pl

Szymański J. (2009).
WordVenture – COOPERATIVE WordNet EDITOR - Architecture for Lexical Semantic Acquisition.
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, pages 28-33
DOI: 10.5220/0002295900280033

ical database. The Cognitive Science Laboratory approach to WordNet development required huge amounts of resources e.g human, time, money (Miller et al., 1990a). With WordVenture, lexical database development becomes common and cheap.

With WordVenture, a user can browse a WordNet dictionary, and display its content on the screen with a graphical user interface based on an interactive graph. It gives a user-friendly way for visualizing very large sets of contextual data. A user can also query WordVenture to find a specified word and display its senses and related concepts. Connections between nodes (words or senses) are illustrated as edges of a given type. To keep graphs clear, a user can set some constraints to visualize only required types of data. There is also the possibility of interactive graph traversing. Selecting one node all elements that are connected with the marked one are displayed (according to given constraints on data selection).

The advantage brought to WordNet development by WordVenture system is a possibility of editing semantic database by the open, Internet community, which fasten lexical data acquisition process. To provide high quality of the acquired data, all changes introduced by users are represented as change propositions, which are approved or rejected by a privileged user – moderator.

3 SYSTEM ARCHITECTURE

It was decided that the WordVenture system will be implemented in client-server architecture, with the following assumptions:

- WordNet database and data access logic resides on the server,
- Data visualization mechanisms reside at the client side and provide interfaces to the lexical database in the form of interactive graphs.

This architecture obliges a developer to implement some functionalities at the server side of the application, but imposes some limitations, especially to communication. The developer has to define communication protocol which will assure flexibility of the data interchange. To widen client-server architecture some elements of the Service Oriented Architecture (SOA) (Erl, 2005) has been introduced. One must meet the following expectations to efficiently implement SOA:

- **Communication Interoperability** – must be assured between different systems and different programming languages. A well-known example of such, is message passing oriented communication

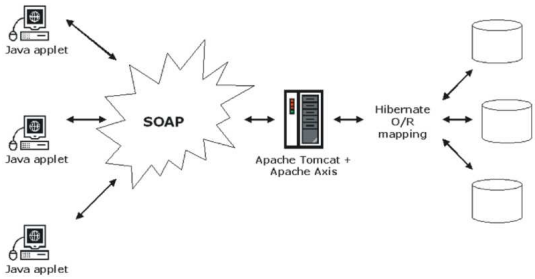

Figure 1: Basic concept of the WordVenture architecture and its elements.

(Palmer et al., 2006). Messages in a defined format are sent between sender and receiver, who performs content-based computations. Neither receiver nor sender have to have precise knowledge about the other's side of a platform.

- **Publishing, Discovery and Service Inquiry** – these are basic concepts of SOA architecture. The three operating sides can be distinguished: **Service Provider** (creates and publishes his services – service producer), **Service Broker** (gives mechanisms to store information about services e.g. physical location of remote service and performs search operations), and **Service Requester** (invokes remote service – service consumer).

One of the most popular implementations of SOA are web services. They've met above requirements, especially communication interoperability between many development platforms e.g. J2EE and Microsoft .NET. Every web service is described in a well-defined and common language – WSDL (Web Service Description Language) (Christensen et al., 2001) and it uses SOAP (Simple Object Access Protocol) as a transport protocol (Scribner et al., 2000). In SOAP, messages are passed as XML documents.

The original implementation of a WordNet database uses text files. Because of their structure, modification is available only with dedicated tools. This type of storage doesn't support synchronous access for modification, nor allows to perform efficiently large amount of queries. It also requires us to create special mechanisms for editing, including synchronization and file structure refactoring, after any operation. To enable editing of a WordNet lexical database we had to perform mappings between WordNet text files and a relational database.

Transformation from text files to its relational representation was performed by the WordNet SQL Builder tool[5].

[5]http://wnsqlbuilder.sourceforge.net

3.1 Server-side Architecture

The server-side of the WordVenture application makes its functionalities available through web services. According to communication interoperability requirement, it is possible to connect client application that can be implemented in different technologies. Web services have been developed and deployed with the Apache Axis framework[6], which resides in the servlets container – Apache Tomcat[7]. Apache Axis framework is a set of libraries and tools which allows a developer to create and publish web services. Axis is just an ordinary web application that can be deployed to any servlet container, especially to Apache Tomcat. It listens for a request from client application that is sent as a SOAP envelope. When a message comes, Axis interprets it and calls a local procedure. Subsequently, a response message is created and sent to the client application. Apache Axis allows programmers to deploy web services such as Plain Old Java Objects (POJOs), which have to have changed extension (from .java to .jws). Deployment can be done by copying jws (Java Web Service) file to proper Axis directory.

The second edition of WordVenture (Szymański, J. and Dusza, K. and Byczkowski, Ł., 2007) introduce mechanisms which allows an moderator to control user actions. To control modifications of lexical database authentication and authorization mechanisms have been created. An anonymous user can only browse the data from WordNet database, if he wants to edit it, he must log in. Every modification introduced to database is represented as ,,change proposition" and is sent to the moderator. This moderator, as a privileged user, can commit or reject every modification proposed by an ordinary user.

Every server functionality allows a user to perform three different groups of actions depending on the role that user has:

- **Functionalities for browsing WordNet Lexical Database** – are available to every user (anonymous and logged-in). After invoking an action on the client-side of application, a proper remote procedure is called on server. The server queries database and sends data to client application. Because of efficiency reasons, all data that is sent between sender and receiver is serialized and compressed, so transmission through the Internet is much more faster.
- **Functionalities that allows a User to edit WordNet Lexical Database** – are only available to registered users. After invoking an edit action on the client-side of an application, the proper change proposition is created. Subsequently, this proposition is sent to the server to be added to database. A privileged user (moderator) can view all change propositions and select commit, other cancel. After committing, a proposition is permanently added to database and can be seen by other users.
- **Administrative Functionalities which are Connected With user Management** – are available only for privileged users – administrators. They can perform user deletion or user rights editing in WordVenture system. Every administrator can give administrative rights to another user.

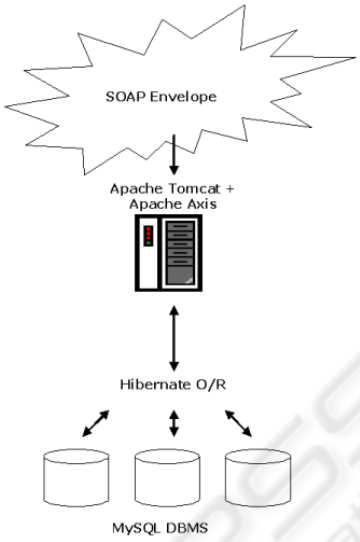

Figure 2: Main view of server-side architecture.

It was decided to use an object-relational mapping mechanism to make our system reusable. Almost all of the object-relational mapping engines use DAO objects (Data Access Object). This project pattern allows the developer to separate data access logic from logic operating on those data. Object-relational mapping mechanisms enable developer to translate data, from relational structure to object structure, what keep proper relations. Each row in a table is translated to a proper object. We've use Hibernate as O/R mapping engine[8] what is a highly-developed and effective solution. One of its main features is the "lazy loading" mechanism (Arendt et al., 1998). It prevents from retrieving at one time all data from database (which can be very inefficient). Lazily loaded mechanisms get data from database only when a end-user wants to see it.

[6]http://ws.apache.org/axis
[7]http://tomcat.apache.org
[8]http://www.hibernate.org

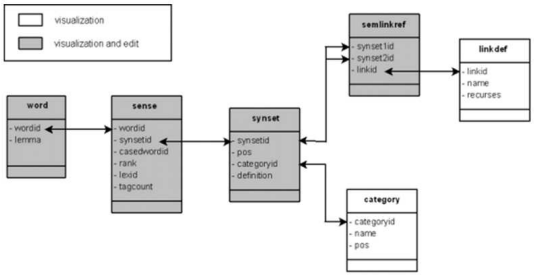

Figure 3: WordNet entities supported by the tool. Grayed out entities have support for both visualization and editing, white entities have only visualization support. Arrows represent relationships between entities.

The Figure 2 presents a detailed diagram of server side architecture, which includes all the abovementioned technological solutions. It shows how the server handles SOAP messages sent by client application. Web service invocation starts when a SOAP envelope comes to the server. Apache Axis framework, resides on Apache Tomcat servlet container, and is responsible for handling SOAP messages. Creating a new web service in Java, from the developers' point of view, requires programming public class with public methods and deploying it to Apache Axis. In a WordVenture system those public classes are used to exploit the Hibernate O/R mapping engine to access database and perform all required queries.

The elements of the WordNet like a word position or morphological definitions are not as much necessary as lemmas and synsets. To simplify the editing process, it was decided to allow only for modification of the semantic net structure. The database structure for handling data provided by WordVenture is presented in the Figure 3, where editable and dictionary tables of the system are shown.

3.2 Client-side Architecture

WordVenture has been developed in rich-client architecture (Boudreau et al., 2007). Because of that, some logic connected with data visualization, can be executed on the client-side of application. Due to ease-of-use requirement, it was decided that client application will be developed as a J2SE 5 applet. The client is a modified TouchGraph component[9] for interactive graph visualization, where graph elements represent WordNet entities. The applet allows a user to:

- **Browse WordNet Lexical Database** – this mechanism is based on a modified TouchGraph engine. It enables the user to navigate over the WordNet semantic network in a user-friendly way. Words and synsets are visualized as graph nodes, connections between them are visualized as graph edges. Additionally, the user can filter graph nodes and edges to obtain required content (according to a selected type), what makes user interface clean and readable.
- **Perform Modifications on WordNet Lexical Database** – the tool enables a user to change graph content by adding, editing, or deleting its elements: nodes and edges. Modification of abovementioned elements of WordNet lexicon (see Figure 3) does not cover all components of WordNet. It only covers the four most desired, from the user point of view, elements of the semantic network: words, synsets, senses and relations, presented in the Figure 3.

Modification of WordNet lexicon is based on well-known rules from other cooperative projects like Wikipedia (Viegas et al., 2007):

- **Changes Patrolling** – every modification of WordNet lexicon is represented as a change proposition that is sent to a privileged user – moderator, who can commit or reject the proposition. This approach is used to trace every activity performed by the cooperative community. Such a mechanism can be used to detect undesirable users' activities: vandalism, violation of copyrights and others.
- **"Free" Character of Wikipedia** – every interested user can join the WordVenture community and cooperate with its creation.

In the WordVenture system a user is able to use the context menu which is available under right click of mouse. Selecting a word or synset makes the system show options available to choose. Functionality of WordNet lexicon editing in cooperative paradigm (Yang et al., 2000) is available only for a logged-in user. In previous release of the system (Szymański, J. and Dusza, K. and Byczkowski, Ł., 2007) synchronous work of many users caused saving only of the last modification. From now on, every modification is saved as a change proposition, and is sent to an moderator. He can choose whether a proposition is permanently saved, or deleted.

Graph-based visualization in a WordVenture system allows a user to work efficiently, and keep clean and readable a large amount of lexical data. In every moment a user can enable or disable required elements of the visualization, which makes his workspace personalized. Additionally, it is possible to zoom in or zoom out view of graph, so a user is able to keep a lot of graph nodes on his workspace.

[9] http://www.touchgraph.com

4 COOPERATIVE APPROACH TO BUILDING LEXICAL NETS

Lack of tools for cooperative editing of semantic dictionary databases is the main barrier for rapid WordNet development. Our mission is to deliver a tool enabling a cooperative editing approach for many users placed in distributed Internet environment. Cooperative editing is connected with publishing the WordNet database and making it open to the Internet community. This brings advantages for faster WordNet development, however some problems may arise:

- **Vandalism** – may cause loss of all important data, kept in current release of lexical database. It also can affect the data structure e.g. creating pointless connections between words and senses. Because of that, it is important to deliver tools which will reduce the risk of the above-mentioned.
- **Simultaneous** work on the same part of database, by many users, may reveal some conflicts resulting from concurrent work of many users at the same time. In the worst case, one user can add connection to an element of the WordNet dictionary that was deleted by another.

The best solution of these above-mentioned problems is to introduce the role of privileged user – moderator. He or she is able to see every change that is proposed to the lexical database dictionary. Every user, after logging in, can edit the lexical database in a restricted way. All introduced modifications are represented as change propositions that are sent to an moderator, who can browse them, and decide whether propositions can be added to database or deleted. All administrative actions result in permanent semantic network update. This approach will also allow us to save history of the database modifications and to detect users – vandals, whose rights can be permanently taken back. According to the basic rule of effective team work („communication, coordination and cooperation") (Kling, 1991), the users were delivered the possibility of continuous communication via a web-based forum. While using it, users can define their own strategies for WordNet lexical database development, reach their own conclusions and also feel all advantages of synergic effect.

The current release of WordVenture system includes all above-mentioned functionalities and is available on project web site: http://wordventure.eti.pg.gda.pl. At present, we are evaluating future proposals for the system, gathering more feedback from users via our web-based forum system, prioritizing future goals, and evaluating the applied solution as a base for a generic approach to semantic data editing tasks.

5 CONCLUSIONS

Our project has been developed and successfully deployed. Currently the WordVenture system has been extended to introduce mechanisms avoiding problems connected with cooperative editing approach:

- Authentication, authorization, and logging users activity – while switching to editing mode, a user is asked to fill-in an authentication form. After logging-in, a user is able to create change propositions that are sent to an moderator. He can trace all the changes and decide whether to approve or reject them.
- Tracing users' change proposals – a privileged user has rights to manage change propositions introduced by other users. Because of that, it is possible to avoid all unwanted changes, and also to review all proposals by a qualified person.

The WordVenture system starts from the newest version of WordNet lexical database (3.0). The system architecture allows us to perform trouble-free actualization of dictionary version with assumption that data structure will not change.

Offering the cooperative editing of the dictionary for the Internet community, seems to be a very attractive way for gathering lexical semantics. It creates the opportunity for fast semantic dictionary development with the cooperation of people from all over the world. It takes down all the duties put on the team while creating the next versions of WordNet dictionary as well. However, we should remember about potential threats which can arise while opening the dictionary for the wide Internet community. The system have been developed based on the experience of Wikipedia. In the current version of the system the risk of vandalism or the unintentional destruction of content has been eliminated, which makes a cooperative approach more reliable.

6 FUTURE MISSION

The WordVenture system has reached the end of its second iteration. In this section we propose changes that will be applied in next versions of WordVenture. Next iteration can include improvements as follows:

1. Internationalization of client-side application – all inscriptions should be organized as supply and should be translated.

2. Integration other lexical networks to WordVenture to make linguistic database richer. By now we consider two projects: Microsoft MindNet (Vanderwende et al., 2005), and ConceptNet (Liu and Singh, 2004)
3. Extension of user activities tracking and edit functionalities to other elements of WordNet database.
4. Introducing improvements to the user interface, especially to administrative part. Currently, an moderator has to manually merge some changes entered by user.
5. Extending the search engine: search by keywords in synset descriptions, etc.
6. Adaptation of user interface to a new displaying engine, which performs more efficient graph visualization, see http://visualwiki.eti.pg.gda.pl/wikiparser.
7. Integration of Wikipedia and WordVenture semantic network.

Future development of WordVenture depends also on users' opinions received via our web based forum. We are waiting for any suggestions and comments about WordVenture – development ideas are welcome. We want to invite everyone to use our system and give us feedback.

ACKNOWLEDGEMENTS

This work was supported by the Polish Ministry of Science and Higher Education under research project N516 035 31/3499.

REFERENCES

Arendt, J., Giangarra, P., Manikundalam, R., Padgett, D., and Phelan, J. (1998). System and method for lazy loading of shared libraries. US Patent 5,708,811.

Boudreau, T., Tulach, J., and Wielenga, G. (2007). Rich client programming: plugging into the netbeans platform.

Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S. (2001). Web services description language (WSDL). *W3C Web Site*.

Erl, T. (2005). *Service-oriented architecture: concepts, technology, and design*. Prentice Hall PTR Upper Saddle River, NJ, USA.

Kling, R. (1991). Cooperation, coordination and control in computer-supported work. *Communications of the ACM*, 34(12):83–88.

Liu, H. and Singh, P. (2004). ConceptNet?a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226.

Miller, G., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. (1990a). Introduction to wordnet: An online lexical database*. *International Journal of lexicography*, 3(4):235–244.

Miller, G., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. (1990b). WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.

Palmer, R., Gopalakrishnan, G., and Kirby, R. (2006). The communication semantics of the message passing interface. *Technical Report UUCS-06-012, The University of Utah*.

Scribner, K., Scribner, K., and Stiver, M. (2000). *Understanding Soap: Simple Object Access Protocol*. Sams Indianapolis, IN, USA.

Szymański, J. and Dusza, K. and Byczkowski, Ł. (2007). Cooperative Editing Approach for Building Wordnet Database. *Proceedings of the XVI International conference on system science*, pages 448–457.

Vanderwende, L., Kacmarcik, G., Suzuki, H., and Menezes, A. (2005). MindNet: an automatically-created lexical resource. *HLT/EMNLP. The Association for Computational Linguistics*.

Viegas, F., Wattenberg, M., Kriss, J., and Van Ham, F. (2007). Talk before you type: Coordination in Wikipedia. In *Hawaii International Conference on System Sciences*, volume 40, page 1298. IEEE.

Yang, Y., Sun, C., Zhang, Y., and Jia, X. (2000). Real time cooperative editing on the Internet. *IEEE Internet Computing*, 4(3):18–25.