

DEBVisDic: Instant Wordnet Building

Adam Rambousek and Aleš Horák

Natural Language Processing Centre
Faculty of Informatics, Masaryk University
Botanická 68a, Brno, Czech Republic

<rambousek@fi.muni.cz, haless@fi.muni.cz>

Abstract

The semantic network editor DEBVisDic has been used by different development teams to create more than 20 national wordnets. The editor was recently redeveloped as a multi-platform web-based application for general semantic networks editing. One of the main advantages, when compared to the previous implementation, lies in the fact that no client-side installation is needed now. Following the successful first phase in building the Open Dutch Wordnet, DEBVisDic was extended with features that allow users to easily create, edit, and share a new (usually national) wordnet without the need of any complicated configuration or advanced technical skills.

The DEBVisDic editor provides advanced features for wordnet browsing, editing, and visualization. Apart from the user-friendly web-based application, DEBVisDic also provides an API interface to integrate the semantic network data into external applications.

1 Introduction

The original wordnet, Princeton WordNet (PWN), is one of the most popular lexical semantic resources in the NLP field (Fellbaum, 1998). The publication of PWN was followed by the multilingual EU projects EuroWordNet 1 and 2 (1998–1999) (Vossen, 1998) and the Balkanet project (2001–2004) (Christodoulakis, 2004) in which wordnets for 13 languages were developed (English, Dutch, Italian, Spanish, French, German, Czech, Estonian, Bulgarian, Greek, Romanian, Serbian and Turkish). In the course of this work, new software tools for browsing and editing wordnets were designed and implemented. Within the

EuroWordNet project the Polaris (and Periscope) tools were implemented and used (Louw, 1998).

For Balkanet project, a new browser and editor VisDic was built at the NLP Laboratory at the Faculty of Informatics, Masaryk University (FI MU) (Horák and Smrž, 2003), since the development of the Polaris tool was closed by 1999.

In comparison with the previous tools, VisDic exploits the XML data format thus making the wordnet-like databases more standard and exchangeable. Not only that, thanks to the XML data format used and to its dictionary specific configurability, VisDic can serve for developing various types of dictionaries, i.e. monolingual, translational, thesauri and multilingually interlinked wordnet-like databases. The experience with the VisDic tool during the Balkanet project has been positive (Horák and Smrž, 2004) and it was used as the main tool with which all Balkanet wordnets were developed.

VisDic, however, has its disadvantages, particularly it was designed for a single user off-line use, and team coordination was really difficult.

2 The DEB platform and the DEBVisDic editor

Based on the experience with VisDic, the team at the NLP Centre FI MU has designed and implemented a universal dictionary writing system that can be exploited in various lexicographic applications to build large lexical databases. The system has been named Dictionary Editor and Browser (further DEB platform) (Horák and Rambousek, 2007) and it has been used in many lexicographic projects so far, among others for the development of the Czech Lexical Database (Rangelova and Králík, 2007), or currently running projects of Pattern Dictionary of English Verbs (Hanks, 2004), and Family names in UK (Hanks et al., 2011).

The DEB platform is based on the client-server architecture, which brings along a lot of benefits.

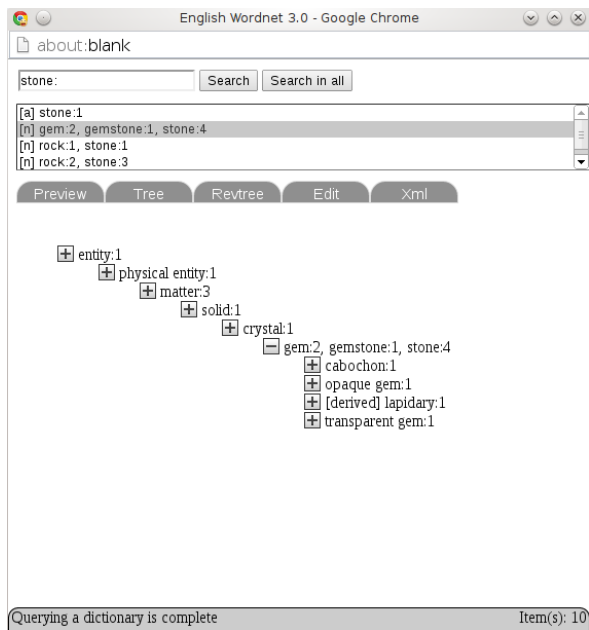


Figure 1: Example of a hypero-hyponymic tree in DEBVisDic.

All the data are stored on the server side and a considerable part of the client-side functionality is also implemented on the server, thus the client application can be very lightweight.

This approach provides very good tools for team cooperation: all data modifications are immediately seen by all involved users. The server also provides well arranged authentication and authorization functions.

The general design of the DEB platform allows to adapt it also for building wordnet-like databases. For this purpose, the original VisDic tool was completely re-implemented on top of the DEB platform, as the current DEBVisDic editor (Horák et al., 2006).

The first version of the DEBVisDic editor was designed as a client application for the corresponding DEB server module, and was implemented using the flexible Mozilla Development Platform (XUL based applications (Boswell et al., 2002)), which was at that time the best option to design and build really cross-platform GUI applications, utilizing open standards.

However, any applications based on the Mozilla Development Platform are limited to the use via Mozilla-based browsers (mainly Firefox) only, while users prefer many different web-browsers. Since the development of DEBVisDic, the Firefox browser has introduced several major changes to the XUL application interface, thus limiting DE-

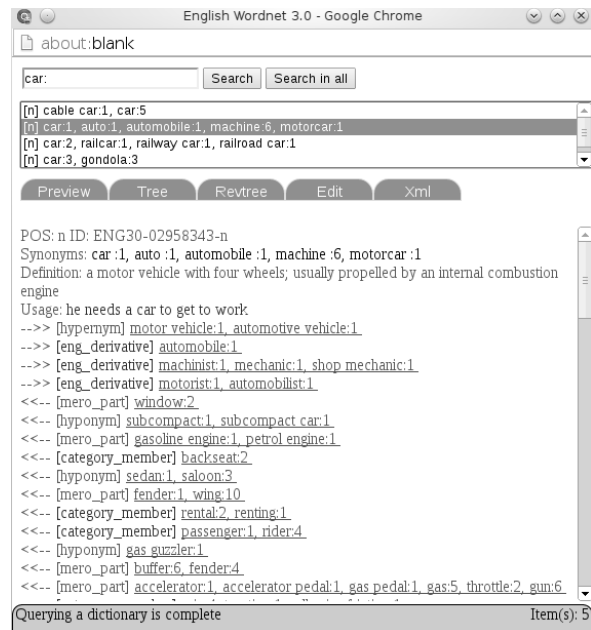


Figure 2: Example of a synset preview.

BVisDic to be used only in specific versions of the Firefox browser. As a result, the editor would need major changes to work with recent Firefox versions.

Fortunately, the current standards for web-based applications support many new features, which are implemented and supported by all the major web browsers. Considering all the options, we decided to re-implement DEBVisDic editor as a general web application, not limited to a single web browser and without the need to install specific browser extensions.

3 DEBVisDic 2

Thanks to the client-server architecture of the DEB platform, no changes were needed on the server side. Only the client side application had to be reimplemented, reusing the existing DEB interface. The main feature requests within the design of the new version were to keep all the DEBVisDic features and to provide an application working in all major web browsers.

As in the previous XUL version, *DEBVisDic 2* (Rambousek and Hrušo, 2013) aims primarily at wordnet-type semantic network browsing and editing, but supports different types of dictionaries. The application consists of a main window with settings and separate windows for each dictionary that the user needs to edit or browser. Single dictionary window includes a list of entries (synsets) and a set of tabbed panels with sev-

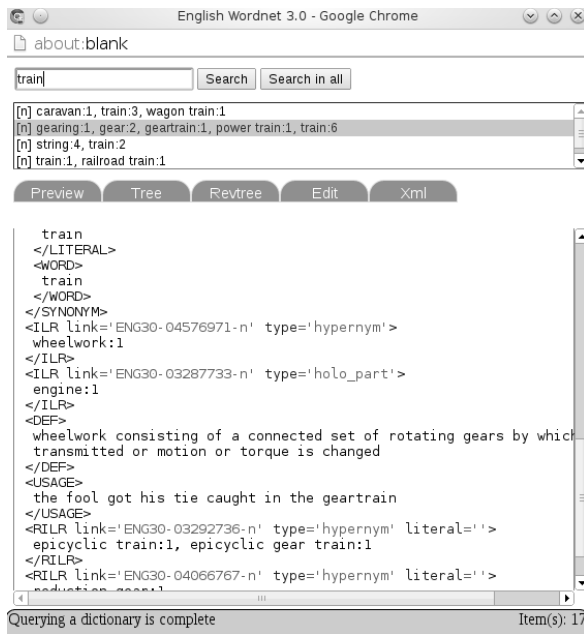


Figure 3: Example of the synset XML representation.

eral types of view on the selected entry: a basic preview, the XML representation, the entry position within the hypero-hyponymic tree, and an editing form. A context pop-up menu (right-click menu) provides functions for displaying and creating inter-dictionary links (e.g. to show the selected synset in another national wordnet or to display all synsets using the selected ontology term).

DEBVisDic 2 utilizes the Model-View-Controller (MVC) architecture and its design follows the MVC principle. Current open standards are used in the application: HTML and CSS for data presentation (view), and JavaScript for application logic scripting (model, controller). The application itself is modular, with separate core shared by all the dictionaries, and a plugin with specific functionality for each dictionary type.

As the implementation of web-related standards (mainly JavaScript) may vary in different browsers, several frameworks and libraries provide a unified environment on top of the browser interface. After reviewing several frameworks, we have decided to use the jQuery library (jQuery Foundation, 2015), which is a versatile JavaScript library for basic document and data manipulation without adding unnecessary features, thus staying lightweight and not slowing down the application.

One of the most challenging features was the implementation of the context menu functions, because of large differences in its implementation

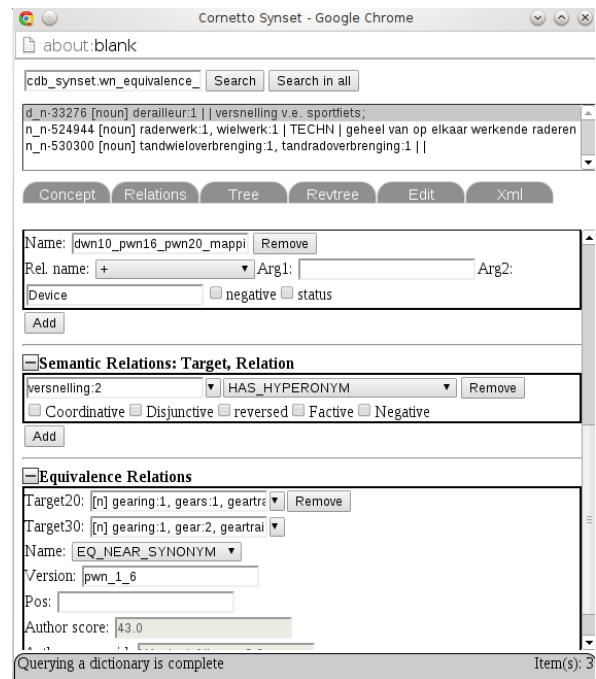


Figure 4: Example of the editing form.

within the main web browsers. In the end, we were able to implement the context menu to keep the same behaviour as in *DEBVisDic* with the help of the jQuery contextMenu plugin.¹ Syntax highlighted view (pretty printing) of an entry in the XML format is provided by the Prettify plugin² (see Figure 3).

Apart from complete reimplementing of the *DEBVisDic* tool, the new version comes with several new features, especially concentrating on team work and complex dictionary editing. For example, *saving user settings* (opened dictionaries and window positions, with the possibility to store more information) *on the server*, thus allowing the user to switch browsers and computers and continue in the work.

Another major new feature is the implementation of *generalized links and relations* between dictionaries. It is possible to use any part of the entry structure (XML-based query) to build inter-dictionary search queries. For instance, selecting all lexical units in a synset, automatically view details of an ontology term for the selected synset, or all synonymic or near synonymic synsets between two wordnet languages.

¹<http://medialize.github.io/jquery-contextMenu/>

²<http://google-code-prettify.googlecode.com/>

username	xrambous	role	manager	e-mail	xrambous@fi.muni.cz	update	reset password
username	scruffy	role	editor	e-mail		update	remove access
username	wloki	role	editor	e-mail		update	remove access
username	dheiko	role	read	e-mail		update	remove access
<input type="checkbox"/> make dictionary public							
<input type="button" value="add user"/>							

Figure 5: User access management.

4 Building new wordnet

To support fast preparation of national wordnets, the DEBVisDic editor was extended with features similar to the DEBWrite application (Rambousek and Horák, 2015). The new application allows any user of the DEBVisDic server to create a new wordnet, without any complicated configuration or a need of advanced technical skills. Right after the straightforward set-up of a new wordnet, users may edit the wordnet data in the DEBVisDic application.

The DEBVisDic application supports copying of synsets from other wordnets (e.g. PWN). Editors have two options: either copy the original synset and translate it to the target language, or to create a new synset and link it to the “pivot” wordnet.³

In the case when the wordnet data are prepared in advance, e.g. in another editor or via “manual” editorial work, it is possible to import a XML file to DEBVisDic. The application also supports an export in the DEBVisDic XML file format.

One of the major assets of the DEBVisDic application lies in its support of team cooperation on the wordnet editing process. DEBVisDic classifies authorized users into one of three possible user roles: a manager, an editor, or a reader (see Figure 5 for an example of user access management).

- The user who created the wordnet is the *manager*. Managers may alter any settings. They may grant access to other users specifying their role. The manager may also decide to make synsets publicly available, which means that no password is needed to browse the semantic network (this *free access* might be regarded as a fourth user role in the dictionary access management).

³This can be either an *interlingual index* (Vossen, 1998) or, which is the most common option today, the Princeton WordNet directly.

- An *editor* may edit synsets before they are set to be published.
- *Readers* may browse and navigate through the published synsets with advanced search capabilities.

5 Data sharing

For sharing the resulting databases and its inclusion to wordnet repositories, e.g. Open Multilingual Wordnet (Bond and Foster, 2013), it is possible to export the wordnet data in several XML formats:

- *DEBVisDic XML*, used as an export/import format in several wordnet editors,
- *Wordnet-LMF* (Soria et al., 2009), developed during the KYOTO project as an extension of the Lexical Markup Framework data model,
- *Lemon RDF* (McCrae et al., 2011), ontology model adopted by the Princeton Wordnet.

To enhance the possibility to share and reuse wordnet resources, DEBVisDic provides a published API (Application Programming Interface) through which the DEBVisDic server functionality can be integrated in external applications, e.g. Visual Browser (Nevěřilová, 2007) for wordnet visualization. Features accessible by the API include complex searching for synsets, extracting various synset data, getting information on the semantic network graph, synset batch editing, or synset translation between several interconnected languages.

6 Conclusions

We have introduced a new web-based version of one of the most widespread wordnet editor, the DEBVisDic application. Besides the easy-to-distribute client application, DEBVisDic now

comes with an extension that allows fast and simple wordnet building. The application is currently in public testing, available at <http://deb.fi.muni.cz/debvisdic>. Future versions will incorporate features based on user feedback, and new options to link separate wordnets together or to shared ontologies. We believe this application will help with creation of new national wordnets, especially for sparsely resourced languages.

In the future, we plan to enhance the DEB-VisDic editor features, both in user experience (e.g. redesign the graphical interface) and format checks to ensure the synset structure follows specified rules.

Acknowledgments

This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarín project LM2010013 and by the national COST-CZ project LD15066.

References

- Francis Bond and Ryan Foster. 2013. Linking and extending an open multilingual wordnet. In *ACL (1)*, pages 1352–1362. The Association for Computer Linguistics.
- David Boswell, Brian King, Ian Oeschger, Pete Collins, and Eric Murphy. 2002. *Creating Applications with Mozilla*. O'Reilly and Associates, Inc., Sebastopol, California.
- D. Christodoulakis. 2004. *Balkanet Final Report*. University of Patras, DBLAB. No. IST-2000-29388.
- Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
- Patrick Hanks, Richard Coates, and Peter McClure. 2011. Methods for Studying the Origins and History of Family Names in Britain. In *Facts and Findings on Personal Names: Some European Examples*, pages 37–58, Uppsala. Acta Academiae Regiae Scientiarum Upsaliensis.
- Patrick Hanks. 2004. Corpus Pattern Analysis. In *Proceedings of the Eleventh EURALEX International Congress*, Lorient, France. Universite de Bretagne-Sud.
- Aleš Horák and Adam Rambousek. 2007. DEB Platform Deployment – Current Applications. In *RASLAN 2007: Recent Advances in Slavonic Natural Language Processing*, pages 3–11, Brno, Czech Republic. Masaryk University.
- Aleš Horák and Pavel Smrž. 2003. VisDic – wordnet browsing and editing tool. In *Proceedings of the Second International WordNet Conference – GWC 2004*, pages 136–141, Brno, Czech Republic. <http://nlp.fi.muni.cz/projekty/visdic/>.
- Aleš Horák and Pavel Smrž. 2004. New features of wordnet editor VisDic. In *Romanian Journal of Information Science and Technology*, volume 7, pages 1–13.
- Aleš Horák, Karel Pala, Adam Rambousek, and Martin Povolný. 2006. DEBVisDic – First Version of New Client-Server Wordnet Browsing and Editing Tool. In *Proceedings of the Third International WordNet Conference - GWC 2006*, pages 325–328, Jeju, South Korea. Masaryk University, Brno.
- jQuery Foundation. 2015. jQuery. <http://jquery.org>.
- M. Louw. 1998. Polaris user's guide. Technical report, Belgium.
- John McCrae, Dennis Spohr, and Philipp Cimiano. 2011. Linking lexical resources and ontologies on the semantic web with lemon. In Grigoris Antoniou, Marko Grobelnik, Elena Simperl, Bijan Parsia, Dimitris Plexousakis, Pieter De Leenheer, and Jeff Pan, editors, *The Semantic Web: Research and Applications*, volume 6643 of *Lecture Notes in Computer Science*, pages 245–259. Springer Berlin Heidelberg.
- Z. Nevěřilová. 2007. The Visual Browser Project. <http://nlp.fi.muni.cz/projects/visualbrowser>.
- Adam Rambousek and Aleš Horák. 2015. DEBWrite: Free Customizable Web-based Dictionary Writing System. In Iztok Kosem, Miloš Jakubíček, Jelena Kallas, and Simon Krek, editors, *Electronic lexicography in the 21st century: linking lexical data in the digital age. Proceedings of the eLex 2015 conference*, pages 443–451. Trojina, Institute for Applied Slovene Studies/Lexical Computing Ltd.
- Adam Rambousek and Tomáš Hrušo. 2013. Web Application for Semantic Network Editing. In *RASLAN 2013: Seventh Workshop on Recent Advances in Slavonic Natural Language Processing*, pages 13–19, Brno, Czech Republic. Tribun EU.
- Albena Rangelova and Jan Králík. 2007. Wider Framework of the Research Plan Creation of a Lexical Database of the Czech Language of the Beginning of the 21st Century. In *Proceedings of the Computer Treatment of Slavic and East European Languages 2007*, pages 209–217, Bratislava, Slovakia.
- C. Soria, M. Monachini, and P. Vossen. 2009. Wordnet-LMF: fleshing out a standardized format for wordnet interoperability. In *Proceedings of IWIC2009*, New York. ACM Press.
- Piek Vossen, editor. 1998. *EuroWordNet: a multilingual database with lexical semantic networks for European Languages*. Kluwer.