

# Online Editor for WordNets

**Borislav Rizov**

Institute for Bulgarian Language  
Bulgarian Academy of Sciences  
boby@dcl.bas.bg

**Tsvetana Dimitrova**

Institute for Bulgarian Language  
Bulgarian Academy of Sciences  
cvetana@dcl.bas.bg

## Abstract

The paper presents an online editor for lexical-semantic databases with relational structure similar to the structure of WordNet – Hydra for Web. It supports functionalities for editing of relational data (including query, creation, change, and linking of relational objects), simultaneous access of multiple user profiles, parallel data visualization and editing of the data on top of single- and parallel mode visualization of the language data.

## 1. Introduction

Hydra for Web is a complex system application for WordNet editing and visualization of complex relational data including parallel data (of two or more wordnets). Due to its complexity, the data encoded into the relational wordnet format need tools that are flexible and easy to use in order to give the users transparent access to data editing.

There are popular user interfaces that allow for the development and visualization of the wordnet relationships as graphs. For example, WordNet Editor (Dusza et al., 2013) is a tool for cooperative development of wordnet database that uses graphical component for graph visualizations with interactive navigation. Unlike the WordNet Editor which follows the idea for a strategy for wikipedia-like editing distributed platform, Hydra for Web integrates an editing functionality and a more simple interface that keeps the structure of a synset with all the relations integrated into one hierarchical structure, and strives for clear visualization of the parallel data.

In our approach to wordnet data, we treat wordnet as a relational structure – consisting of a set of objects and a set of binary relations between them. The objects are of three types – Synset, Literal and Note. The Literals (i.e., the words) in one synset are connected with it via a relation called *literal*. The Notes objects represent the textual data in wordnet – usage examples and notes. Every usage example is connected to its synset by the relation *usage* in a way similar to literals.

## 2. Viewer

Hydra for Web <sup>1</sup>, is a single page web application that uses as backend the API of the open source modal logic tool for wordnet development Hydra (it is freely distributed; (Rizov, 2008), for the browser functionalities see also (Rizov and Dimitrova, 2008). <sup>2</sup>.

The search system provides results in all of the available languages (selected by the user) – the database currently contains (open source) wordnets for 23 languages. The navigation bar has a drop-down menu for switching between the wordnets to be worked with. Except for the default Princeton WordNet 3.0, Bulgarian wordnet (BulNet) and Romanian wordnet (RoWN), the selectional options of which are visualized in corresponding pairs, the user can enable additional wordnets or disable others by means of a modal dialog.

The interface is currently available in English, Bulgarian, and Romanian.

---

<sup>1</sup>Available at <http://dcl.bas.bg/bulnet/>

<sup>2</sup>Hydra for Web is freely downloadable at: <http://dcl.bas.bg/hydra/>

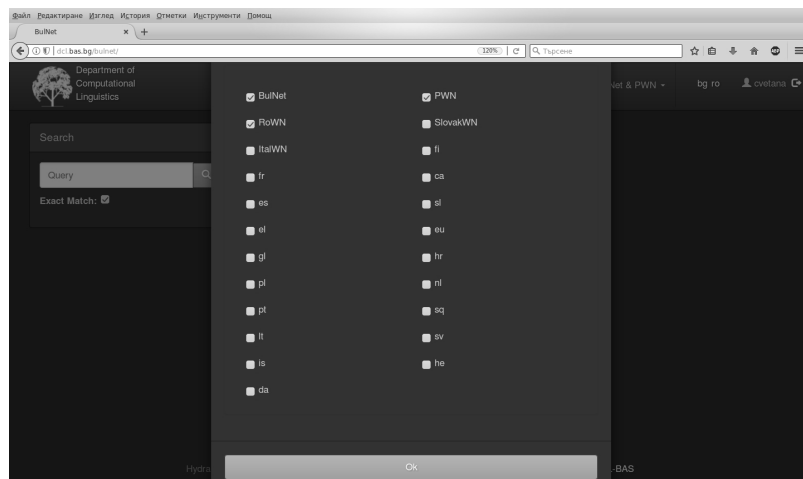


Figure 1: Hydra for Web – WordNet as a Multilingual Dictionary

The tool allows for searching into databases of different language wordnets with a single query as shown on Fig. 2 which illustrated how the word ‘**aspirin**’ can be searched for in Greek and Finnish.

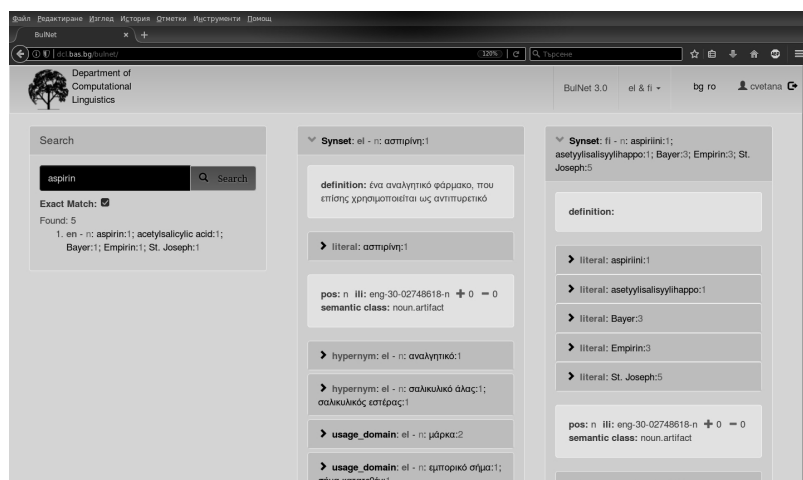


Figure 2: Hydra for Web – WordNet as a Multilingual Dictionary

The selected result – the first one from the list below the Search pane – is propagated to the right hand side visualizer(s). Hydra for Web supports two visualization modes: single mode, and bilingual mode (as on Fig. 2) where you see the correspondences of the selected synset (‘**aspirin**’ in English) in the mode’s languages (in Greek and in Finnish – to the left and to the right, respectively).

Every object visualisation is recursive in a sense that every relation (hypernym, holo\_part, etc.) that leads to other object (i.e., synset) is expandable in the same way as the root one. The data in objects like pos, ILL, etc., are available immediately, while the relations are loaded by means of AJAX query, but asynchronously without blocking the UI.

## 2.1. Search

The tool allows searching for an exact match of a word string – a single word such as [**aspirin**], or a multiword unit, e.g., [**aspirin powder**] as in Fig. 3, or a non-exact match search which returns any synset where the searched word is found.

Although the three types of object in our approach to wordnet are fully-fledged, the search panel returns all the synsets that contain a literal matching the search query.

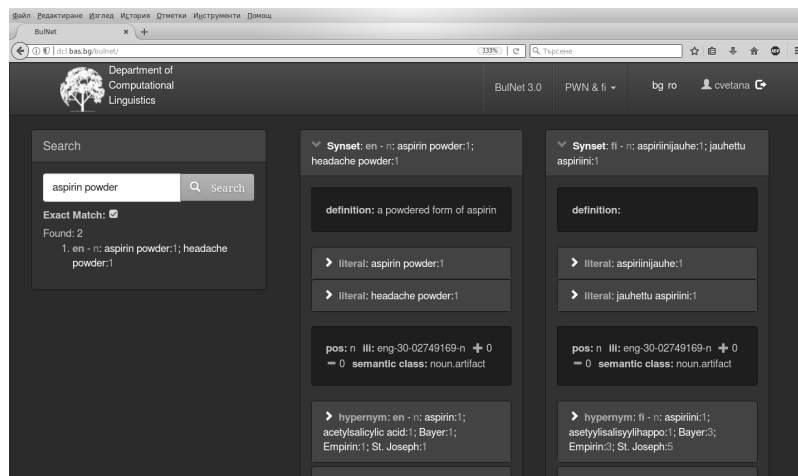


Figure 3: Hydra for Web – Multiword Unit

The search input is enhanced with autocomplete (with prefix match) as shown on Fig. 4 where synsets with the string ‘**powde**’ are shown while typing the word **powder**.

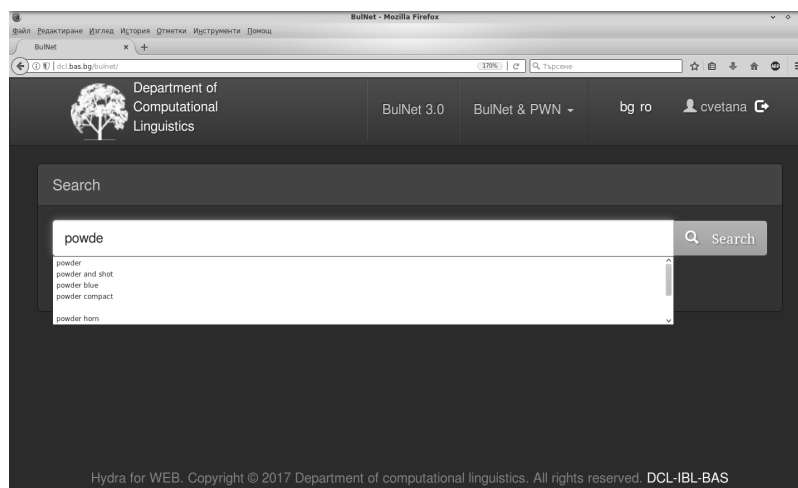


Figure 4: Hydra for web – autocomplete

The search returns a paginated list with the respective synsets in the database with the results shown at once being limited to 30 synsets in a list below the Search input but the user can navigate by using the button Next and Previous to browse between the pages with results.

To limit the results shown, the search respects word (string) boundaries, i.e., the user can search only for whole words but not parts of the words.

In the next section, we will present the editing functionalities of the tool.

## 2.2. Editing

Users can work on any wordnet (language) once it is added in the database.

The online editor Hydra for Web allows to: edit object’s data (some of the fields require free text like definition, while others are with predefined value list – f.ex. the part-of-speech); add object (literals and notes are added by button clicks in the parent objects); delete object; add binary relation between existing objects.

The editing functionalities of the tool will be illustrated by showing the process of editing the synset [kaysiya, kaysievo darvo] in the Bulgarian wordnet. This is automatically generated synset which has

not been validated yet as Bulgarian wordnet has been expanded by automatically translating a number of synsets (literals, definitions, and usage examples) to be further edited by experts.

When the user searches for a word (literal), the tool shows its status – literals that have been validated by an expert, are visualised in the standard color (white in Fig. 5), while those that have not been validated yet, are dimmed (muted).

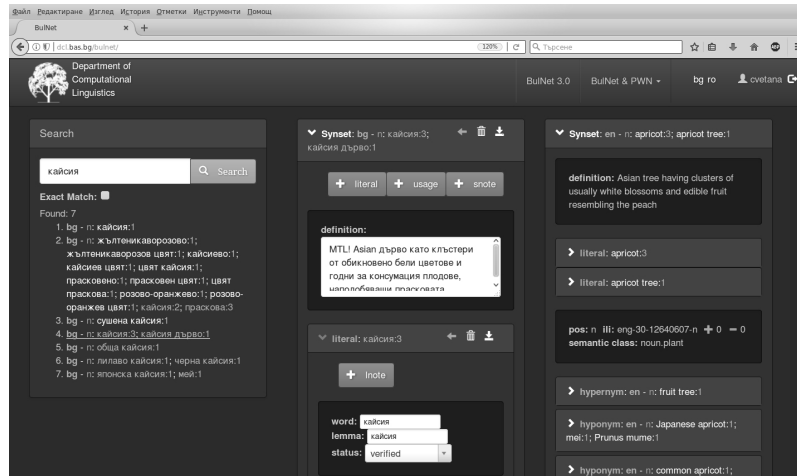


Figure 5: Hydra for Web – online editing

A synset can be edited by clicking on the top right-edge Edit button of the panel to put the linguistic unit (Synset, Literal or Note) panel in Edit mode – the data visualisation controls are replaced with those for editing.

The Edit panel for a synset consists of subpanels for the elements in the synset which are at least four: the set of literals constituting a synset; the definition; the literals visualised as a list one below the other (each literal can be edited as an independent object as shown on Fig. 5); and information for the current synset only – part-of-speech (pos), ILLI, sentiment values according to SentiWordNet, semantic prime.

From top to bottom, the following elements are part of the editor panel (for Synset object) as shown on Fig. 5:

1. Panel header – textual representation of the synset – all the literals to the left, followed by buttons for canceling (the arrow sign), deleting (the ‘bin’ sign), and saving the synset.
2. Three buttons for adding (with the plus sign) [literal], [usage] and [snote] relations of the synset.
3. The definition.
4. The literals – each can be edited independently by clicking on the Edit button and opening an Edit panel which is much like the Editor panel of the parent synset. By clicking on the literal – without opening the Edit panel – the user can view the whole information about the literal at hand (word, lemma, status, and [lnote] plus the entire synset it pertains to below).
5. Information about: pos, ILLI, sentiment values according to SentiWordNet, and semantic class. All values of these categories are editable – pos, SentiWordNet values, and semantic class are available as a list with fixed values (these are shown on the English synsets to the right in Fig. 5).

Other elements are: usage ((snote)), relations such as hypernym, hyponyms, derivational relations, morphosemantic relations, and others (see the example with the verb ‘cook’ on Fig. 6 below).

The synsets to which a currently edited synset is linked to via a relation (hypernym, hyponym, etc.) are given as a list after the subpanel (5) and each of the linked synsets can be edited further on its own.

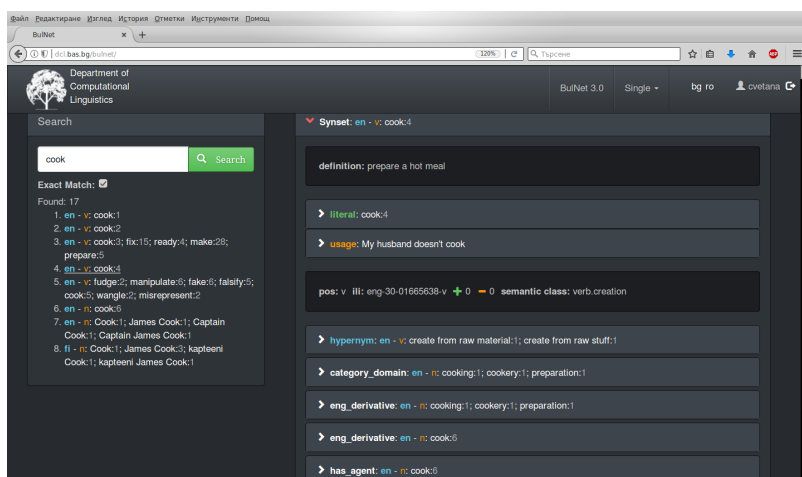


Figure 6: Hydra for web – whole synset, single mode

### 2.3. Linking

The Linguistic Units can be connected by introducing a relation between the two connected LUs. It is accomplished by means of a Wizard. To start it, the user clicks on the Connect button to the left of the Edit button on the unit panel. The procedure requires the following steps:

Step 1: A new Select Relation panel is opened to replace the Search panel. The new panel offers a list of all the relations available for the selected type of LU.

Step 2: The target LU of the relation is shown via a Search panel identical to the main Search panel. The search returns a list of synsets to be linked to the selected synset. Fig. 7 shows a selection of the  $\langle is\_agent\_of \rangle$  relation that has to link the synset [kovach:1; zhelezar:1] ‘blacksmith’ to the synset [kova:2] ‘forge, hammer’. The selection of a target synset from the searched for list in the Search panel shows the whole synset below the list in the Search panel. If this is the intended synset, the user clicks on the button Connect and the link is visualized on the panel to right.

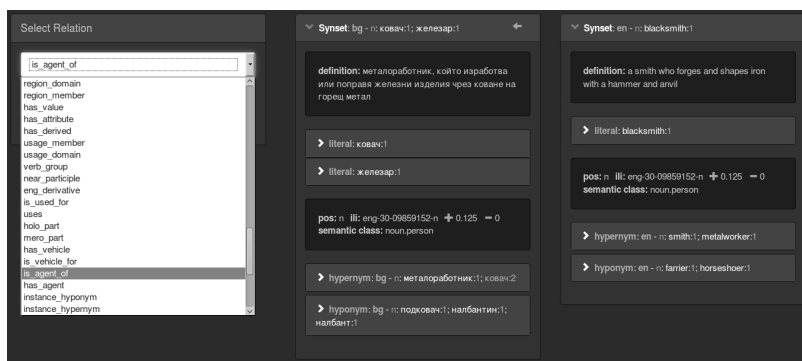


Figure 7: Hydra for web – selection of elements

### 3. Concurrent Editing

All modified data is propagated to the other connected users immediately by means of notifications by the wordnet server. In case of a conflict (when, for example, the same object is edited by more than one user), the last user is responsible for merging the data. When receiving a notification that some data is in edit mode, Hydra puts it in merge mode. If there is a possibility for not receiving a notification (i.e., there is a network problem), different strategy can be implemented.

Every client (web page instance) stores locally the copy of the data to be edited, and this copy is

submitted to the server alongside with its modifications. The server detects the conflict if any, and notifies the client, which puts the data in merge mode.

#### 4. Users

Hydra for Web is freely accessible to all. Anonymous users can view (search and browse) the language data in the 23 wordnets in the database (but cannot edit the data – this option is available only to users with specific privileges). Additionally, the system is enhanced with user management with the following privilege options for every given language/wordnet:

- None: The wordnet is unavailable to the user.
- View: The user can search and browse this wordnet.
- Edit: The user can edit the data and relations in this wordnet.

Hydra for Web is mobile-friendly on a small width (mobile) (exemplified on Fig. 8 with the Romanian synset **agă:1**), where the panels are ordered successively.

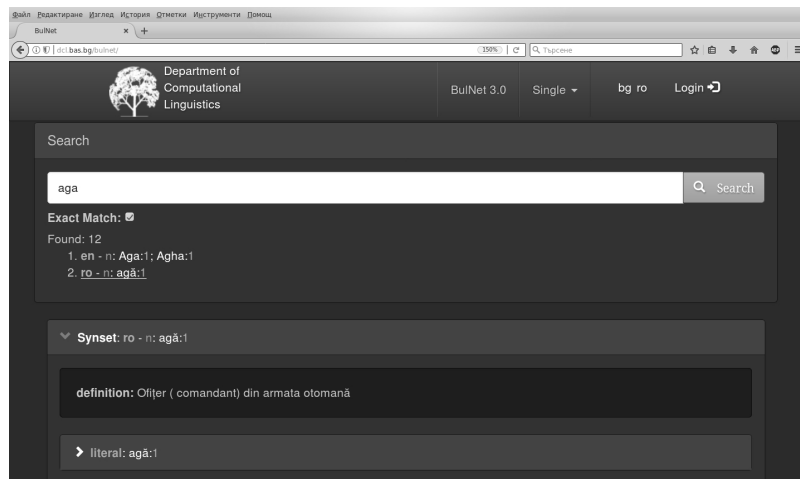


Figure 8: Hydra for web – selection of elements

#### 5. Implementation

Hydra for Web is implemented by means of modern web technologies and libraries. Its source code is relatively small, straightforward and easy to maintain and extend *Hydra for web* with new features.

Hydra for Web is built with Node.js<sup>3</sup> and Express<sup>4</sup>. It is a single page application and uses one of the most popular HTML, CSS and JS frameworks – Bootstrap<sup>5</sup>.

Hydra for Web is themed in Slate from Bootswatch<sup>6</sup>. Bootstrap makes easy the GUI to be responsive, and so it is mobile friendly.

For the html rendering, the very clean and elegant PUG template engine<sup>7</sup> (formerly known as Jade) is used.

Many of the tasks in the GUI are solved in the client with the use of Knockout.js<sup>8</sup> framework. It uses declarative bindings, dependency tracking and provides automatic UI refresh.

<sup>3</sup>Node.js® is a JavaScript runtime: <https://nodejs.org/>

<sup>4</sup>Web application framework for Node.js <http://expressjs.com/>

<sup>5</sup><http://getbootstrap.com/>

<sup>6</sup><https://bootswatch.com/>

<sup>7</sup><http://jade-lang.com/>

<sup>8</sup><http://knockoutjs.com/>

The wordnet data retrieval is made by means of the Wordnet Service. The retrieval uses AJAX and is completely asynchronous (non-blocking).

All the notifications of edited data are propagated to the the other users with SockJS which gives a coherent, cross-browser, Javascript API which creates a low latency, full duplex, cross-domain communication channel between the browser and the web server.

## **References**

- Fellbaum, C. (1999). *WordNet: an Electronic Lexical Database*. Cambridge, MA: MIT Press.
- Borislav Rizov. 2008. *Hydra: A Modal Logic Tool for Wordnet Development, Validation and Exploration.. Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, ELRA.
- Borislav Rizov and Tsvetana Dimitrova. 2014. *Hydra for Web: A Browser for Easy Access to Wordnets.. Proceedings of the Global Wordnet Conference*, Tartu, pages 339–343.
- Dusza, K., Byczkowski, L., and Szymanski, J. (2007). Cooperative editing approach for building wordnet database. *Proceedings of the XVI international conference on system science*, 448–457.
- Miller, G. (1995). Wordnet: A Lexical Database for English. *Communications of the ACM*, November 1995, 38(11): 39–41.
- Vossen, P. (2002). WordNet, EuroWordNet and Global WordNet. *Revue française de linguistique appliquée*, 1/2002 (Vol. VII), 27–38.