

New Features of Wordnet Editor VisDic

Aleš HORÁK, Pavel SMRŽ

Faculty of Informatics, Masaryk University Brno
Botanická 68a, 602 00 Brno, Czech Republic

E-mail: (hales,smrz)@fi.muni.cz

Abstract. This paper deals with wordnet development tools. It presents a designed and developed system for lexical database editing, which is currently employed in many national wordnet building projects. We discuss basic features of the tool as well as more elaborate functions that facilitate linguistic work in multilingual environment.

1. Introduction

Princeton WordNet became one of the most popular language resources. It is currently used in many areas of natural language processing such as information retrieval, automatic summarization, document categorization, question answering, machine translation etc. To integrate into the applications, many researchers work with the Princeton database and transform data to their own proprietary formats.

The Princeton team also developed a data browser for WordNet which can be downloaded together with English data from the web page <http://www.cogsci.princeton.edu/~wn/> both for Windows and UNIX platform. No WordNet editing tools are provided as the only instruments for majority of the lexicographic work in Princeton are standard text editors. The consistency of data is not therefore checked during the editing process itself, it is postponed to later phases.

Year by year the number of Princeton WordNet clones and WordNet-inspired initiatives increased. In 1998–1999 the EU project EuroWordNet 1 and 2 [1] took place, in which multilingual approach has dominated and WordNets for 8 European languages, particularly for English, Dutch, Italian, Spanish, French, German, Czech and Estonian, have been developed. The Interlingual Index (ILI), Top Ontology, set of Base Concepts and set of Internal Language Relations have been introduced as well [2]. These changes also led to the design and development of the new database engine for EuroWordNet and it resulted in the editing and browsing tool called Polaris [3].

In 2001 the EU project Balkanet [4] has been launched which can be viewed as a continuation of EuroWordNet project. It has been conceived as a multilingual as well and within its framework WordNets for 6 languages are being presently developed, particularly for Greek, Turkish, Romanian, Bulgarian, Serbian and Czech. Before Balkanet has started it had already been obvious that Polaris tool had no future because its development had been closed and as a licensed software product (by Lernout and Hauspie) it had been rather expensive for most of the research institutions involved (typically universities). Moreover, the system had been provided only for MS Windows platform.

As the developers of Czech WordNet within EuroWordNet 2 project we came to the conclusion that a new tool for WordNet browsing and editing has to be developed rather quickly. At the same time we realized that it was necessary to look for the solution that would also support establishing the necessary standards for WordNet like lexical (knowledge) databases. Thus we decided to develop a new tool for WordNets based on XML data format, which can be used for lexical databases of various sorts. The tool is called VisDic and it has been implemented recently in Natural Language Processing Laboratory at Faculty of Informatics, Masaryk University for both Windows and Linux platform [5, 6].

2. Basic Functionality

VisDic was developed as a tool for presentation and editing (primarily WordNet-like) dictionary databases stored in XML format. Most of the program behavior and the dictionary design can be configured. With these capabilities, we can adopt VisDic to various dictionary types—monolingual, translational, thesaurus or generally linked wordnet lexicons.

2.1. Multiple Views of Multiple Wordnets

The main working window is divided into several dictionary panels. Each panel represents a place for entering queries and browsing context of one specified wordnet dictionary. The panels can display different wordnets as well as multiple contexts of the same dictionary.

The contents of a panel offers, besides the query input and matching results list, a set of overlapping notebooks tabs each of which represents one kind of view of the same entry from the list of results. The order, the type and even the content of each notebook tab is specified by the user in the configuration files (see 4). The main types of views are described in the following sections.

2.2. Freely Defined Text Views

The content of the Text View notebook tab is entirely built from the user definition that follows the XML structure of the wordnet entry. The editor can thus present an easily readable view of the entry with highlighting important parts of the entry content (see Fig. 1).

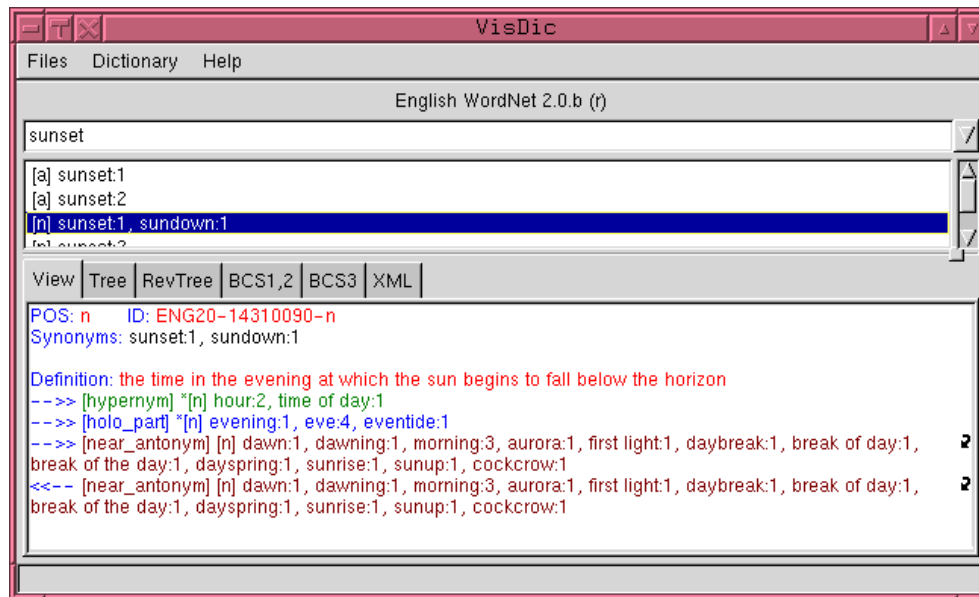


Figure 1. An example of freely defined text view of wordnet entry

2.3. Edit

The editing capabilities allow to give the user a full control over the content and linking of each entry in the wordnet hierarchy. To prevent the user from moving the entry as an object in the spider web of the linkage relations, the linguist rather specifies all the links in a textual dialog, where all the bindings are displayed in one place with consistency checks after each change request.

The actual contents of the Edit notebook tab is also entirely driven by the user instructions in the configuration, where each editing field is named and assigned to an XML tag in the entry.

2.4. Tree and RevTree

The wordnet dictionaries are specific by a heavy network of various kinds of relations between the dictionary entries with the function to capture the ontology relations on the underlying natural language.

The navigation in such environment is thus a crucial point of a successful linguistic work with wordnet data. Since the linkage relations generally do not need to obey any rules, that could make the resulting structure to be an arbitrary directed acyclic graph, or DAG. VisDic implements a browsing mechanism for general graphs. The navigation process works with two interconnected notebook tabs, which always both start at the same dictionary entry and display its position in the graph represented as a breadth-first path trees of all the linkage relations that lead from the entry to other entries in the dictionary. Each of the notebook tabs displays mutually opposite

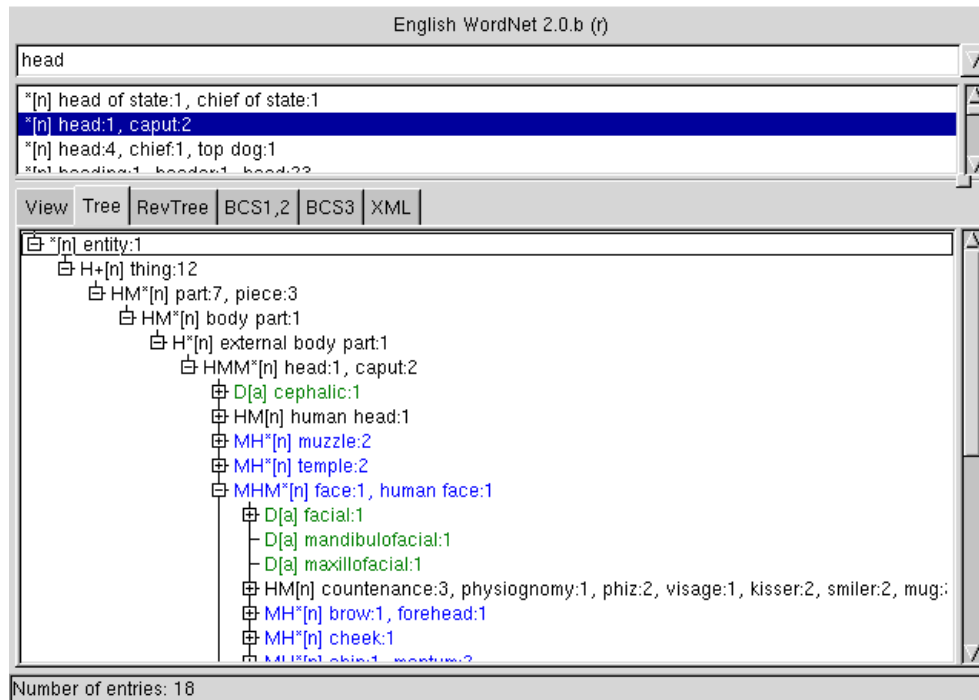


Figure 2. The tree-like navigation in the wordnet linkage relations graph

linkage relations, allowing the user to choose the direction of graph navigation in every step.

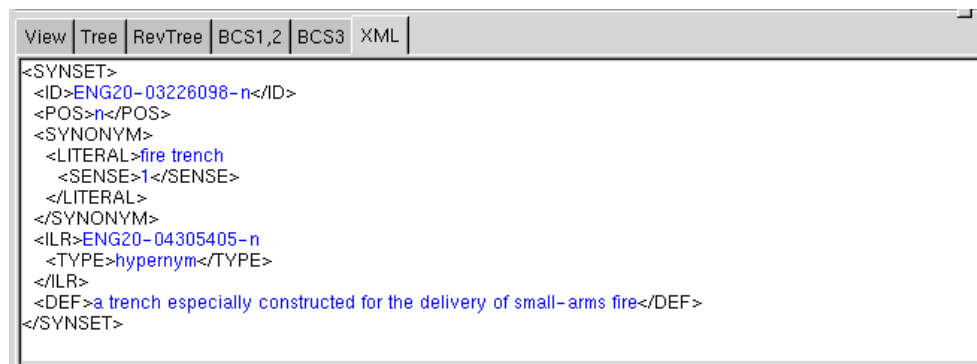
To facilitate the orientation and to help to position the entry in the wordnet hierarchy, the navigation also displays the path from the entry to its top in the hyperhyponymical relation tree (see Fig. 2). For more advanced navigation the linguist may also use advanced tree browsing techniques (described in 3.3.).

2.5. Query Result and External File Lists

Common actions in the wordnet creation and editing often include processing of a subset of entries based on certain criteria. VisDic offers a suitable kind of views for this situation, which allow to prepare a notebook tab with a list of entries matching any user specified query or a list of entries identified by entry-IDs gathered in a plain text file.

2.6. Plain XML View

Sometimes users need a thorough view into the data contained in the dictionary entry. XML View notebook tab offers this possibility. In this view, the user can see a graphically structured XML text, which represents the entry structure as it is stored in the dictionary (see Fig. 3).



```

<SYNSET>
  <ID>ENG20-03226098-n</ID>
  <POS>n</POS>
  <SYNONYM>
    <LITERAL>fire trench
      <SENSE>1</SENSE>
    </LITERAL>
  </SYNONYM>
  <ILR>ENG20-04305405-n
    <TYPE>hypernym</TYPE>
  </ILR>
  <DEF>a trench especially constructed for the delivery of small-arms fire</DEF>
</SYNSET>

```

Figure 3. Raw XML view of a synset entry

3. Advanced Functionality

The basic functionality described in the previous section generally conforms to any XML based dictionary. However, linguistic work specialized to wordnet creation and editing requires some more specific and more sophisticated functions in the editor.

3.1. Synchronization

Within the creation of a national (e.g. Czech) wordnet, which would correspond to the English wordnet as a primary reference, one of the most frequent operation is a lookup of a dictionary entry (synset) from one wordnet in another dictionary. Such lookup uses either the SYNSET.ID tag (as a direct equivalent) or one of the, so called, equivalence tags (or attributes) defined in the configuration. An example of such tag may be REVMAP or MAPHINT used to help the linguist to process ambiguous link references between various versions of English wordnet.

The lookup function in VisDic can work in two modes: as an instant (one time) lookup — the *Show (by)* operation, and also as a firmly established link between two notebook tabs called the *AutoLookUp (by)*. In case of *AutoLookUp*, any move to another dictionary entry in the source notebook tab leads to an automatic lookup of the new entry in the destination tab. VisDic allows to have any acceptable combination of autolookups among all the notebook tabs.

3.2. Editing Support

The efforts of unifying national wordnets based on the English wordnet in many cases lead to copying of synset information between different language dictionaries. Such functionality in VisDic is split into two common situation — either the SYNSET.ID of an existing synset is to be unified with the ID of the English synset (*Take key from* operation) or a whole new entry is to be copied to another dictionary (*Copy entry to*).

3.3. Tree Browsing

The basic navigation in related synsets (in some cases reduced to the hyper- and hyponymical relations tree) is supplemented with two important wordnet operations — *Topmost entries* and *Full expansion*.

The Topmost entries operation identifies all synsets, which are (in the tree subset of linkage relations) found as the roots of relational hierarchy, i.e. are not hung below some other synset. This helps the linguist to identify the level 1 entries as well as so far unfiled entries.

The Full expansion allows the user to see all possible descendants of a selected synset in the linkage relations graph. During the operation cycle detection techniques check the violations of tree properties in the graph. Some relations can be also configured to be left out from the full expansion process.

3.4. Consistency Checks

Semi-automatic processing, which often takes part in the national wordnets creation, as well as common human processing of the data inevitably brings in the possibility of mistakes. The inconsistencies, which may be revealed as a duplicity, are controlled by VisDic consistency checks, which contain

- check duplicate IDs
- check duplicate literals and senses
- check duplicate synset literals
- check duplicate synset links

These checks allow the linguist to identify the most common errors e.g. after merging data from various sources.

3.5. Journaling

The work on a large and representative national wordnet usually employs more than one linguist working on the data. The synchronization of the resulting dictionary is made possible in VisDic with the usage of *journaling*.

During the work with VisDic, any changed to the data is marked in a journal file. Each journal file is specific to one dictionary and one user at a time. Such journal file can then be “applied” to the dictionary data and merged with the original. In this way, the simultaneous work of several linguists can be easily interchanged with a common data source.

4. XML configuration

Most of the functionality in the VisDic wordnet editor can be adopted to the local needs by means of its configuration files. All settings for the VisDic application are stored in several XML files.

4.1. Global Configuration

The main configuration file (`visdic.cfg`) serves for global application data storage such as the list of dictionaries, the list of views, fonts, colors or query history. All information is stored in XML structure.

The first-level subsections of the global configuration are:

colors In the `COLOR` section the user can define colors which are then referenced by its name in dictionary configuration files. Each color is enclosed in its name tag and consists of three hexadecimal values separated by commas, representing consequently its red, green and blue components. Each value can be in range `<0x0000,0xffff>`.

fonts The `FONT` section defines fonts which can be referenced by the defined names from dictionary configurations. Each font definition is enclosed in its name tag and its value correspond to the font string description.

application settings The `APPL` section contains all global data that are related to the application state. The most common settings, that can be found here, are:

- `DICT` – path to a dictionary that is presented in the list offered to the user.
- `OPEN` – relative number of a dictionary that should be opened in one notebook tab.
- `AUTOLOOKUP` – definition of a synchronization link between two notebook tabs.
- `SIZE` – size of the notebook tab in percentage of the main window width.
- `HIST` – history of last queries that were entered by the user in a specific notebook entry line.

A shortened example of a global configuration file is displayed in Fig. 4.

4.2. Dictionary Specific Configuration

Each wordnet dictionary has its special configuration file (`dictionary.cfg`), which enables the linguist to set up most of the texts displayed in the application as well as the content of notebook tabs specific to the particular dictionary with respect to the XML structure of the entries.

The configuration contains attribute settings of the dictionary and sections describing the layout of the dictionary views. The main attributes in the dictionary configuration are:

- `NAME` – full name of the dictionary. This name is presented to the user in various places in the application, e.g. on the top of the dictionary notebook tab.
- `SHORT_NAME` – short name of the dictionary.
- `MAIN_TAG` – the default XML tag in the user queries (e.g. `SYNSET.SYNONYM.LITERAL`).

```

<?xml version="1.0"?>
<CONFIG>Visdic general configuration file
  <COLOR>Colors definition
    <BLACK>0x0000, 0x0000, 0x0000</BLACK>
    <WHITE>0xffff, 0xffff, 0xffff</WHITE>
    <RED>0xffff, 0x0000, 0x0000</RED>
    <GREEN>0x0000, 0xffff, 0x0000</GREEN>
  ..
</COLOR>
  <FONT>Fonts definition</FONT>
  <APPL>
    <DICT>/nlp/wn/visdic/data/eng20/wneng20</DICT>
    <DICT>/nlp/wn/visdic/data/eng171/wneng171</DICT>
    <DICT>/nlp/wn/visdic/data/eng15/wneng15</DICT>
  ..
  <AUTOLOOKUP>v2-v1</AUTOLOOKUP>
  <OPEN>1
    <SIZE>43</SIZE>
    <HIST>
      <LINE>trench</LINE>
      <LINE>house</LINE>
      <LINE>dog</LINE>
    ..
  </HIST>
</OPEN>
  <OPEN>5
    <SIZE>57</SIZE>
    <HIST>
      <LINE>pes</LINE>
    ..
  </HIST>
</OPEN>
</APPL>
</CONFIG>

```

Figure 4. The global configuration example (... stands for shortened parts)

- **MAX_QUERY** – limit of the number of results of a query.
- **MAX_VIEW** – limit of the number of characters displayed in the user defined text view.
- **CHARSET** – name of a character set indicating the encoding of the dictionary. This information is necessary for correct manipulation with the dictionary in some systems.

The rest of the dictionary configuration file contains sections defining the list of the available dictionary views and their content or the list of duplicate checking actions in the application menu.

4.2.1. Visual Definitions

The **VISUAL** section describes the way, how to display dictionary entries. Definitions are enclosed in tags corresponding to their names. VisDic uses primarily two special visual definitions. The first is called **VISDIC_SHORT** and it presents the entry in a short one-line format (e.g. list of all entries matching the query or within a tree view). The second visual definition, named **VISDIC**, describes the content of the user defined text view, i.e. it presents the entry in a more descriptive way.

Every tag in the XML file can be displayed its own way. The definition contains C-like string format specifications consisting of a string in double quotation marks and other parameters. These parameters have the following meaning:

- %c – color name (taken from visdic.cfg), changes the current color.
- %f – font name (taken from visdic.cfg), changes the current font.
- %s – string, it can be @tag:name, a tag name, or @tag:value, a tag value.
- %i – includes the output of subtags' processing.
- %K+ – in the tree view, stop expanding the view under the current entry.
- %K – in the tree view, delete the current line from the tree.

The format string can include parts that are displayed only under a certain condition. The available conditions are

- \\{^...\\} – display ... only if the tag is the first in the list.
- \\{\$...\\} – display ... only if the tag is the last in the list.
- \\{*...\\} – display ... only if the tag is not the last in the list.

An example of usage of the conditional parts of the format string can be a comma separated list of literals with their senses:

```
<SYNONYM>"%i"
  <LITERAL>"%s:%i\\{* , \\}",@tag:value
    <SENSE>"%s",@tag:value</SENSE>
  </LITERAL>
</SYNONYM>
```

The visual definition of each XML tag can contain a test for the value of the tag in the dictionary entry. For instance, the various type of wordnet relations between synsets can be transcribed in colored one-letter acronyms like this

```
<ILR>"%i"
  <TYPE>="hypernym": "%cH",BLACK</TYPE>
  <TYPE>="holo_member": "%cM",BLUE</TYPE>
  <TYPE>="derived": "%cD",DARK_GREEN</TYPE>
  <TYPE>"%c[%s]",RED,@tag:value</TYPE>
</ILR>
```

a special tag named DEFAULT stands for any tag and it is used for tags that do not have their own definitions.

4.2.2 Views

The VIEW section specifies the design of notebook tabs. Each tab is described by one LIST subsection. Each tab has its own name in the NAME tag and its own type in the TYPE tag. According to the type, the LIST subsection can include other specifications of the tab content:

- XML view has no other options. It just displays the XML structure of a dictionary entry.

- **USER** type has **DEF** tag referencing the visual definition of the user defined text view.
- **TREE** type contains two tags specifying parent and child link tags in the dictionary and the **DEF** tag for the visual definition used in the presented tree-like ordering of entries.
- **EDIT** type describes the form fields for editing one dictionary entry. The subsection contains **ITEM** or **BUTTON** tags. Items refer to XML tags in **TAG**, each has its own head label in **HEAD** and its own item type in **TYPE**. The appearance of the form field is specified in the **EDIT** tag. It can be a single line entry (**ENTRY**), a multi line entry (**TEXT**) or a checkbox (**CHECKBOX**). All form fields used for editing the link or reverse link tags (see 4.3.) will be displayed as combo boxes with an arrow ↗ on the right side of the box, which allows the user to navigate to the referred entry. All form fields that represent a tag which can occur more than once are supplemented by two buttons ➕ and ➖. These buttons are used for adding another instance or removing the current instance of the tag.

The **BUTTON** tags define buttons that run one of the storage actions. Each button has its label specified in the **TEXT** tag and its type in the **TYPE** tag. The type can be either **NEW** for creating the new entry, **DELETE** for deleting the current entry or **UPDATE** for saving the edited entry content.

- **WORD** type view presents a list of all words from the dictionary that can be found among values of the given tag.
- **ENTR** type view is a list of entries that meet a condition given by the user query in the **QUERY** tag.

4.2.3 Main Menu Actions

The **MENU** section describes a list of dictionary-specific actions which can be added to the VisDic main menu. All these actions will be appended to the *Dictionary* submenu.

An example of the actions that can be specified in the **MENU** are the duplicity checking actions. These actions are looking for duplicate values within the dictionary, either among entries or within a single entry. The action definition is enclosed in the **DUPL** tag. The **TYPE** tag chooses the kind of comparison – **ENTR** for comparing entries or **ITEM** for comparing items within the range of one entry. The **NAME** contains a name of the action, which will be displayed in the menu. The **TAGS** tag enlists all tags that are included in the duplicity checking, the tags are separated with the | sign. If a tag begins with a dot (.), then the tag is considered as a subtag of the previous tag.

Examples of the duplicity checking actions are:

- searching for all entries (synsets in WordNet) having the same **SYNSET**. **ILI** value

```

<DUPL>
  <TYPE>ENTR</TYPE>
  <NAME>Check duplicate ILI numbers</NAME>
  <TAGS>SYNSESET.ILI</TAGS>
</DUPL>

```

- identification of all pairs *literal:sense* in WordNet stored in more than one synset. Here, the `.SENSE` tag corresponds to `SYNSESET.SYNONYM.LITERAL.SENSE` subtag of the `SYNSESET.SYNONYM.LITERAL` tag

```

<DUPL>
  <TYPE>ENTR</TYPE>
  <NAME>Check duplicate literals & senses</NAME>
  <TAGS>SYNSESET.SYNONYM.LITERAL|.SENSE</TAGS>
</DUPL>

```

- finding all literals in WordNet that occur more than once in one entry (synset)

```

<DUPL>
  <TYPE>ITEM</TYPE>
  <NAME>Check duplicate synset literals</NAME>
  <TAGS>SYNSESET.SYNONYM.LITERAL</TAGS>
</DUPL>

```

4.3. Dictionary definition

Each dictionary has associated, besides its configuration file, a definition file named *dictionary.def*. This file describes the XML structure of the dictionary. The structure of the definition file contains features that are specific to the wordnet-like XML dictionaries.

The definition file format is a plain text with each row corresponding to one XML tag. The line format is

```
level tag min max type args
```

where the corresponding fields contain

- *level* – the tag level (0 for the top level).
- *tag* – the tag name.
- *min* – minimal number of occurrences of the tag within its supertag.
- *max* – maximum number of occurrences of the tag within its supertag (–1 means infinite number).

0	SYNSET	1	1	N	
1	ID	1	1	K	
1	POS	1	1	N	
1	SYNONYM	1	1	N	
2	LITERAL	1	-1	N	
3	SENSE	1	1	I	@maxbyparent+1
3	LNOTE	0	1	N	
1	ILR	0	-1	L	
2	TYPE	1	1	N	
1	RILR	0	-1	R	SYNSET.ILR
1	BCS	0	1	N	
1	DEF	0	1	N	
1	USAGE	0	-1	N	
1	SNOTE	0	-1	N	
1	STAMP	0	1	N	

Figure 5. An example of a dictionary definition file.

- *type* – the kind of the tag. It can be one of
 - N – normal text entry.
 - I – integer number entry. In the *args* column a function for the default value can be stated.
 - K – key value uniquely identifying the entry. Such key is used by the following L, R, and E kinds of tags.
 - L – link to another synset, it represents a semantic relation.
 - R – similar to L. It is defined as reversed link specified in the *args* column.
 - E – contains an external information stored in another dictionary. The name of the external tag and the path to the dictionary are contained in the *args* columns. The path is absolute or relative to the VisDic initial directory, not relative to the dictionary path.
- *args* – extra arguments for some kinds of tags.

An example of a dictionary definition file can be found in Fig. 5.

5. Conclusions and Future Directions

VisDic, during its rather short history, has already proved its suitability for lexical database creation. The main power of VisDic manifests itself especially in development of highly interlinked databases such as wordnet. Its unique features have assured VisDic the leading role in many wordnet editing projects.

The development of such tool is never really closed. The future directions of our work will concentrate at specific support for linguists, improvements in the customization and user interface and team cooperation functionality. Entirely new horizons ap-

pear in the ongoing development of VisDic successor, the client-server lexical database editor DEB [7].

Acknowledgement. This work was supported by Ministry of Education of the Czech Republic Research Intent CEZ:J07/98:143300003 and by EU IST-2000-29388.

References

- [1] Eurowordnet project website, <http://www.illc.uva.nl/EuroWordNet/>.
- [2] VOSSSEN, Piek, editor, *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*, Kluwer Academic Publishers, Dordrecht, 1998.
- [3] POLARIS, Louw M., *User's guide*. Technical report, Belgium, 1998.
- [4] Balkanet project website, <http://www.ceid.upatras.gr/Balkanet/>.
- [5] PAVELEK, T., PALA, K., VisDic – A New Tool for WordNet Editing, In *Proceedings of the First International Global WordNet Conference*, Mysore, India. Central Institute of Indian Languages, 2002.
- [6] HORÁK, A., SMRŽ, P., Visdic – Wordnet Browsing and Editing Tool, In *Proceedings of the Second International WordNet Conference – GWC 2004*, Brno, Czech Republic. Masaryk University, 2003.
- [7] SMRŽ, P., POVOLNÝ, M., Deb - dictionary editing and browsing, *Proceedings of the EACL03 Workshop on Language Technology and the Semantic Web: The 3rd Workshop on NLP and XML (NLPXML-2003)*, pages 49–55, Budapest, Hungary, 2003.