Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 469–476
ISBN 978-83-60810-27-9
ISSN 1896-7094

WordnetLoom: a Graph-based Visual Wordnet Development Framework

Maciej Piasecki, Michał Marcińczuk, Adam Musiał, Radosław Ramocki, Marek Maziarz
Institute of Informatics, Wrocław University of Technology, Poland
Email: {maciej.piasecki,michal.marcinczuk}@pwr.wroc.pl

***Abstract*—The paper presents WordnetLoom – a new version of an application supporting the development of the Polish wordnet called plWordNet. The primary user interface of WordnetLoom is a graph-based, graphical, active presentation of wordnet structure. Linguist can directly work on the structure of synsets linked by relation links. The new version is compared with the previous one in order to show the lines of development and to illustrate the introduced difference. A new version of WordnetWeaver – a tool supporting semi-automated expansion of wordnet is also presented. The new version is based on the same user interface as WordnetLoom, utilises all types of wordnet relations and is tightly integrated with the rest of the wordnet editor. The role of the system in the wordnet development process, as well as experience from its application, are discussed. A set of WWW-based tools supporting coordination of team work and verification is also presented.**

I. An Editor for Wordnet

A LARGE wordnet[1] is a very complex graph of many thousands vertices and arcs, where vertices represent lexical units and sets of lexical units, and arcs represent lexico-semantic relations of several types. A wordnet can be expressed in a simple formal language, cf the language defined for the needs of Princeton WordNet [3] and used in "lexical source files". However, the size and the complexity of a large wordnet makes manual encoding of the wordnet structure error prone. Error probability can be reduced by the introduction of specialised wordnet editors that free linguists from learning a formal language of description and provide some forms of error checking. Several wordnet editors have been proposed.

Two applications were constructed for the EuroWordNet project [4]: Polaris — an editing tool, and Periscope — a graphical database viewer. However, Polaris is a commercial tool, its development has been closed, and it is not commonly available for research. Moreover, it is a modification of a previous ontology editor, is tightly associated with the EuroWordNet structure and was constructed for a limited number of platforms.

Because of the Polaris limitations, a new tool called VisDic was created for the Czech WordNet [5]. In VisDic the relation definitions are still written in text windows, but an XML based format is used and some immediate browsing is possible in the tool, e.g. bi-directional browsing of graphs of semantic relations. VisDic was used in the Slovene Wordnet [6] and was available for research. VisDic was a monolithic application directly working on XML files, contrary to its direct descendant DEBVisDic [7] – a client-server, lexical database editor. DEBVisDic reimplements and extends the functionality of VisDic and is based on the client-server architecture and an XML database server. DEBVisDic has become a popular tool used in several projects.

Both tools are oriented on editing a wordnet synchronized with wordnets for other languages by the Interlingua Index [4], that complicates their basic structure and user interface.

For the needs of the GermaNet[2] development a dedicated wordnet editor called GernEdiT was constructed [8]. GernEdiT stores data in a relational database and provides concurrent access for many users at the same time. GernEdiT offers graph-based, graphical presentation of the wordnet structure, but limited to the hypernymic links only. Other types of relations, including relations between synsets (called *conceptual relations* in GermaNet) are not shown. Moreover, direct editing of the relation graph is not possible. Data must be first changed or added via dialogue forms. This makes the association between an action and the effect more remote from the user point of view. Direct editing of the graph is more compatible with the idea of GUI (Graphical User Interface).

The majority of existing tools for visualising a wordnet focus only on the hypernymy structure, e.g. [9], [10], and only some of the graph-based tools offer possibilities of editing, e.g. [11].

When the project on the construction of plWordNet[3] started in 2005, DEBVisDic [7] was not available and VisDic [5] was not an alternative, as it is a single machine tool. We needed a tool supporting a group of linguists working in a distributed environment. A dedicated wordnet editor called plWordNetApp was built. It was not as universal and flexible as DEBVisDic, but it implemented the assumed procedure of linguistic work. Its interface was designed according to the estimated ways of performing linguistic tasks. plWordNetApp enabled cooperation of a linguist team on the basis of a central database accessed via Internet. A brief characteristic of the editor is given in Sec. II.

plWordNetApp has been used by the Polish wordnet development team since the year 2006. During this period we

[1] A wordnet is a large electronic thesaurus following the basic construction features of Princeton WordNet [1], [2]. In WordNet, one and multi-word lexical units are grouped into *synsets* – "sets of near synonyms", that are linked by semantic relations derived from lexico-semantic relations.

[2] GermaNet is a wordnet for German, the second largest wordnet in the world.

[3] The first publicly available wordnet for Polish – Polish name: *Słowosieć*

collected rich experience concerning usability drawbacks of the editor. Editing and browsing complex graphs of hypernymy appeared to be the biggest problem. The problem was gradually rising with the increasing complexity of the plWordNet hypernymy structure[4]. Linguists requested visualisation of the hypernymy graphs which would be improved in the comparison to simple tree-based, static visualisation offered by plWordNetApp. Thus an improved version of the editor was constructed with the graphical presentation of the wordnet structure as a basis for all kinds of interaction.

The goal of this paper is to present a new wordnet editor called WordnetLoom, which is built around the idea of visual, direct editing of the wordnet graph structure. The editor is integrated with a tool called Wordnet Weaver, which delivers automatic support for extending the wordnet.

In the rest of the paper we will first briefly describe plWordNetApp – the basis for the present version of the application. Next the core graph-based view will be introduced. Full integration of WordnetWeaver with the rest of the application is discussed. The paper is concluded with the presentation of a set of web pages enabling browsing of data important for the coordinators of a linguistic group and plans for further development.

II. plWordNet 1.0 Development Tools

From the very beginning of designing plWordNetApp we were trying to find a solution for the necessity of presenting and editing complex wordnet structures and, at the same time, fitting the resulting complex structures of user interface onto one screen. In order to avoid cluttering the main screen, we decided to divide the user interface of the application into two main screens, presenting two main perspectives:

- the *perspective of lexical units*[5]
- and the *perspective of synsets*.

plWordNetApp GUI lets the lexicographers avoid the use of an artificial language for the description of semantic relations, starting with introduction of a new lexical unit (LU) and its description; this improves on the practice in WordNet and GermaNet [1], [15]. Both browsing and making decisions (during editing) are done via GUI screen controls and transparently recorded in the server database.

[4]The hypernymy is a lexico-semantic relation between a lexical unit of a more general meaning and a semantically subordinated lexical unit of a more specific meaning – a relation similar to the superclass-subclass relation – e.g. 'plane figure' is a hypernym of 'tree diagram'.

[5]Informally, lexical unit is a semantically disambiguated word in a broad sense. Technically, a *lexical unit* is a pair: *lemma* and one of the *meanings* represented across different occurrences of this lemma in language utterances, cf [12], [13], e.g. a pair: the lemma *kolejka*, [polysemous] 'narrow-gauge railway', 'train', 'round' or 'queue', and its meaning represented in *plWordNet* 1.0 [14] by the symbol `kolejka 3` – a kind of railway, means of transport.
A *lemma* is a morphological word form selected as an exponent of the whole set of word forms of the same Part of Speech, such that all word forms from the set are described by the same shared grammatical categories and the word forms differ only in the values of the subsequent categories, e.g. *program* 'program' as an exponent of a set: *programu*, *programie*, *programach*, ... or *maszyna parowa* 'steam engine' (multiword lemma) as an exponent of the set: *maszyny parowej*, *maszynie parowej*, *maszyn parowych* ...; we assume that the choice of a particular word form as a lemma is arbitrary and constrained only by some tradition.

The perspective of lexical units, cf [13, pp. 39], was intended to support grouping lexical units into synsets. The lexical unit list, present in the system, can be filtered according to several criteria, e.g. a selected domain. To facilitate searching, each synset is also automatically assigned to the domain of its first lexical unit. Domains are the main tool in organising the work of the linguist team: linguists are assigned complete domains to process by the coordinator. Once a lexical unit has been selected, its properties are presented and can be edited. A new lexical unit can be added in the lexical unit perspective (but in some other parts of plWNApp GUI, too).

All synsets including a given lexical unit are shown in the tabbed panel below the lexical unit property panel, cf [13, pp. 41]. Editing of the selected synset is possible in the tabbed panel to the right of it — the second, hidden tab pane contains the synset properties. We assumed that first a selected lexical unit is assigned to an existing synset or freshly created one, and then the synset is edited.

In the hidden tab pane of the synset list panel, one can browse and edit a list of lexical relations[6] of the selected lexical unit, i.e. between pairs of lexical units, e.g. antonymy or derivational relations. Derivational relations play an important role in the structure of plWordNet [16].

In order to support the consistency among the linguists' decisions a *substitution test* defined for the given relation is presented to the user, prior to adding any new instance of a semantic relation. The test templates are defined and can be edited by coordinators in a dedicated window. On the basis of a template, a test instance is generated from the template by filling it with the inflection forms of the tested lexical units — the morphological analyser *Morfeusz* [17] was applied. The inflection form properties are specified in templates by IPIC codes. The mechanism of the tests makes plWordNetApp different from other wordnet editing tools, e.g. DEBVisDIC.

From the property panel of the selected synset, the user can switch to the synset perspective set to this synset as the *source synset*.

The five panels of this synset perspective, cf [13, pp. 41], can be divided into the following groups:

- selection and editing of a *source synset* – the synset for which we are going to define a relation or whose relations we are going to browse and edit,
- selection and editing of a *target synset* of a relation to be defined.
- browsing of existing relations.

There are two possible views of synset relations: a tabular one, cf [13, pp. 41] and a tree one (the hidden tab pane). According to the linguists' demands, the initial browsing facility was extended with editing synset relations directly in this view. The browsing panel also enables the navigation along the graph of relations. The possibility of editing synsets directly in this perspective was introduced in order to facilitate

[6]By lexical relation we mean a lexico-semantic relation described by pair of lexical units, in contrary to synset relations that link synsets and originate from the lexical relations defined for the lexical units belonging to them.

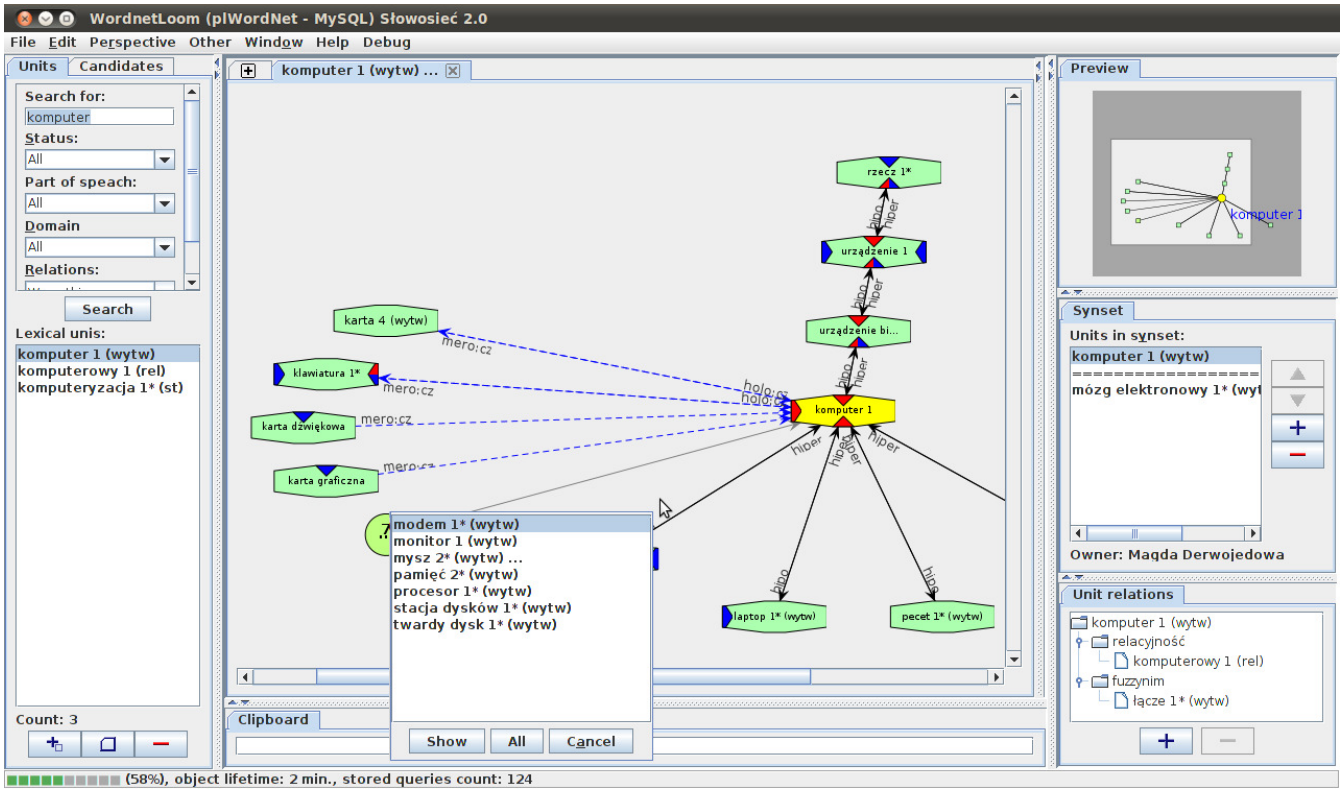


Fig. 1. Screenshot of the active graph-based presentation of the wordnet structure (Glosses for synsets: *rzecz* 'thing', *urządzenie* 'device', *urządzenie biurowe* 'office device', *komputer* 'computer', *karta* 'card', *karta dźwiękowa/graficzna* 'audio/graphic card', *klawiatura* 'keyboard' *modem* 'modem', *monitor* 'monitor', *mysz* 'mouse', *pamięć* 'memory', *procesor* 'processor', *stacja dysków* 'disk drive', *twardy dysk* 'hard disk').

the correction of the initial synsets, e.g. it is possible to extract some lexical units from the source synset and to create a new hypernym synset.

Whenever a user wants to introduce an instance of a synset relation, the appropriate substitution test is presented. According to the definition of the synset relation introduced in plWN, it must be valid for any pair of lexical units from both synsets: the source and the target one. Thus, lexical semantic relations are extrapolated from relations on lexical units to relations on synsets. The substitution test window facilitates selecting all possible pairs of lexical units (from both synsets) and generating instances of the test.

Besides the main screen of the synset perspective, cf [13, pp. 41], an additional screen of synset editing was introduced during the plWordNetApp development. This screen is used for browsing synsets and have a layout similar to the lexical unit perspective: a large list of synsets on the left (rich filtering possibilities), and the tabular view of the selected synset relations, plus all synset editing panels on the right. The screen and the lexical unit perspective are synchronised, i.e. the filter setting and the selected synset and/or lexical unit are transferred to and from the lexical unit perspective during switching. It facilitates browsing the lexical unit relations of the lexical units belonging to the given synset.

III. Graph-based Wordnet Editor

The construction of plWordNetApp was focused on its use on slower computers as were used in 2005 and effectiveness of the primarily keyboard-based interaction. A new version of the application, called *WordnetLoom*[7], is built around the active, graphical presentation of the wordnet structure as a central element of the user interface. The presentation is described as 'active', as it enables not only browsing but also full editing of the structure. An example screenshot is presented in Fig. 1.

The organisation of the presentation is a mixture of the often used radial layout and a vertical tree-like scheme. The latter is often used for presenting large graphs and also wordnets, e.g. [9]–[11]. In the radial layout one synset (node of a graph) is selected as the central one (the present location of the user focus) and other synsets are presented around it in circular layers defined by the number of links from the centre.

According to our analysis, the radial layout expresses several drawbacks from the point of view of the active wordnet presentation. There is only one central element and simultaneous presentation of several sub-graphs sharing some nodes on one screen is difficult. A situation in which there are several

[7]The name refers to a static role played by the wordnet editor – it is a tool, which enables manual construction of a wordnet, while WordnetWeaver is a semi-automatic tool, which is a kind of 'active participant' of the wordnet development process.

paths from the central node to a node shared by the sub-graphs causes deformation of the layout. This is a problem for any layout, but the radial one is especially prone to this issue.

We modified this scheme in order to preserve presentation of the hypernymy in the form of a tree-like scheme. Thus, a synset which is selected by the user becomes the synset in focus. Along the vertical dimensions its hypernymy (direct and indirect) and hyponyms are presented, respectively above and below the central synset.

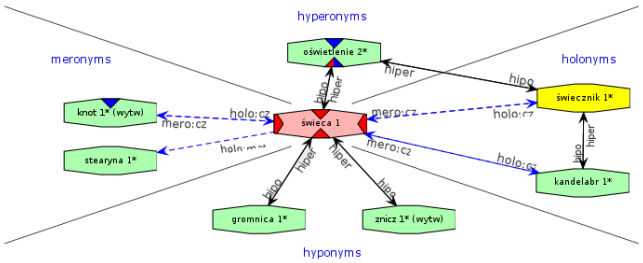


Fig. 2. Layout of the presentation of relation links (Glosses for synsets: *oświetlenie* 'lighting', *knot* 'wick', *świeca* 'candle', *kandelabr* 'candelabra', *stearyna* 'stearin', *znicz* 'candle', *świecznik* 'candlestick').

Meronymy and holonymy are shown along the horizontal dimensions: meronyms to the left and holonyms to the right, respectively. An example of the layout of relation presentation is shown in Fig. 2. Lines dividing the space and the relation names written in a blue bold face font are not a part of the screen, but they have been added for the presentation purpose only. Other types of synset relations, like near-synonymy[8], dweller (e.g. *mieszczanin* 'burgher' – *miasto* 'city'), markedness[9] are presented below the central synset, as they are close in their character to the hyponymy. Relations are visually differentiated by line style, line colour and labels. The definition of labels follows symbols used in the source files of Princeton WordNet. Optionally, shorten names of relations can be used instead, as it is shown in Fig.2.

Interaction starts with searching for lexical units in the left panel. Next, after a lexical unit has been selected the corresponding synset is shown as the central synset. Initially only synsets linked by the direct links to the central one are presented. However, the user can navigate by pressing triangular buttons on the sides of the octagonal synset symbols. The colour of the triangular buttons signals whether none, some or all links of the given type have been opened for the given synset.

A synset can contain several lexical units[10], each lexical unit can participate in several lexical relations, a synset can have several dozen relation links and there can be many synsets – it is impossible to show legibly all this information on one computer screen at the same time. Thus we had to remove a lot of data from the main panel and leave only the minimum of information required for the general understanding of the structure on it. We left only one lexical unit presented per synset, this is compatible with a linguistic practice of assigning the lexical unit which is the most representative for the given synset to first position. Moreover, in the case of a large wordnet the vast majority of synsets include only one lexical unit.

The whole set of lexical units of the central synset can be browsed and edited on the `Synset` panel – the pane in the middle of the right most column in Fig. 1.

Lexical relations have been moved to the separate pane in the bottom-right corner – `Lexical relations` in Fig. 1. Only relations for a lexical unit which is selected in the `Synset` panel are presented. Keeping the detailed information concerning synsets and lexical units in the separate panels defers access to this information and increases the load of the user's short-term memory, but it is a compromise with the possibility of having a more broad view on the wordnet structure.

Synsets are assigned a minimal representation in the graph, but still a wordnet structure of relations can be quite dense in some parts, and many synsets located in the upper parts of the hypernymy structure can have several dozens or even hundreds[11] of outgoing hyponymic links. Thus, by default, WordnetLoom presents only up to $k = 5$ (a parameter) links of the given synset relation (presently only hyponymy). If the number of links is greater, than only the first $k - 1$ are automatically presented and the rest is folded and hidden under the symbol of a circle with the number of hidden synsets shown on it. After clicking the circle a small list-window is shown, the user can browse across hidden links and unfold any sub-group of them – this situation is presented near the bottom of the main panel in Fig. 1. In any moment a link can be folded back into the group of the hidden links.

In large graphs the user can quickly get lost seeing only a small portion of a graph. As a support for navigating in the space of a large graph we introduced a mini-map overview panel in the top-right corner in Fig. 1. Synsets are presented only as point-size circles, but colours are used to distinguish the central synset and selected synsets. Moreover, synset descriptions are given in tooltips shown in response to holding the mouse pointer over some circle.

Visual, graph presentation gives an opportunity to perform editing directly on the relation structure. The context and potential consequences of an action to the presented part of the wordnet structure are clearly visible on the screen. However, only a subgraph can be presented and the biggest problem was to find a way for adding new relation links from synsets presented on the screen to those not. A gradual enlargement of the subgraph being unfolded by clicking triangular buttons is always possible, but only to some space limits. Unfolding of huge graphs is not an option if one has to look for the target

[8]Near-synonymy is different than synonymy expressed by synsets, and is used to link lexical units of the very close meaning but belonging to the different language registers, e.g. *chłopiec* 'boy' – *gówniarz* 'squirt'.

[9]Markedness is relation originating from the regular derivational associations and encompasses several sub-types like dimunitives, augmentative, expressive forms etc.

[10]There are a few synsets which include about 20 LUs each.

[11]E.g. *czynność* 'activity' has 376 hyponyms, mostly gerunds.

synset of a link in some different part of the wordnet. In order to facilitate simultaneous browsing of different parts of the wordnet the central panel was implemented as a multi-panel window enabling opening each sub-graph in a new sub-panel of the central panel. However, switching between sub-panels complicates the way in which selection of a target synset is performed. The user has to remember the state of the user interface as "in selecting a target". In order to make both the source and the target of the action accessible without switching between sub-panels, we introduced a *clipboard* – and additional panel – in which the user can keep references to the synsets that he is working on, are interesting to him, etc. `Clipboard` is located below the main panel, see Fig. 1. When adding a new relation link the user first selects the source synset by right clicking it, and then selects the target synset from the active sub-panel or from the clipboard. Only synset symbols are shown in the clipboard but one can activate a sub-panel, which is already open, or open a new sub-panel for a synset in the clipboard with mouse double clicking on its symbol.

The functionality of simultaneous opening several graphs in separate sub-panels supports comparing different parts of the wordnet structure – a challenging task but very important in the case of a large wordnet developed by a team of several linguists. Erroneously separated parts of the hypernymy structure can be immediately linked, and other corrections can be introduced, too.

It is worth to emphasise here that the graph-based editing and presentation is a significant advantage offered by *WordnetLoom* in comparison to *plWordnetApp*, at least according to the opinions of the linguist team members. The most important improvement is that a linguist can work on all relations (linking synsets and lexical units) using one set of panels without the necessity of switching to other screens. Moreover, he can still follow the general wordnet structure as it is defined by the synset relations and can be perceived only from the perspective of groups of inter-linked synsets not only singular synsets.

IV. Integrated Semi-automated Support

WordnetLoom includes a new version of WordnetWeaver as its integrated part [13]. WordnetWeaver is a tool supporting semi-automated expansion of plWordNet. It consists of the two parts:

- the algorithm of Activation-Area Attachment (henceforth, AAA) which for a new lemma from the outside of plWordNet generates a set of suggested new lexical units on the basis of several knowledge sources,
- and an user interface based on the active presentation of hypernymy subgraphs as descriptions of the suggested relations.

The latest version of the AAA algorithm was described in details in [18], here only a brief characteristics is given below. AAA utilises several knowledge sources describing semantic association of lemma pairs. The used set of five knowledge sources includes: Measure of Semantic Relatedness (1), which has been constructed automatically on the basis of a large corpus, lemma pairs – possibly hypernymic – extracted from the corpus by hand-written patterns (2) and also patterns learned automatically (3). Moreover, lemma pairs for which the measure returns high values are filtered by a classifier (4), which has been trained for the recognition of wordnet relations and finally lemma pairs in which both lemmas are mutually high on the their lists of the mutually close semantically related according to the measure (5), cf [13].

AAA works in three steps, described below in the perspective of adding one new lemma to plWordNet.

1) Semantic fit between a new lemma and each lemma in the wordnet is calculated on the basis of weighted voting applied to information concerning the new lemma from the knowledge sources.
2) Semantic fit between the new lemma and each synset is calculated:
   a) on the basis of cumulated fit between the new lemma and the synset lemmas,
   b) and also on the basis of the fit between the new lemma and synsets that are located in the close distance to the given synset.
3) Connected hypernymic sub-graphs of synsets expressing higher semantic fit to the new lemma are identified and presented as descriptions of potential new lexical units for the new lemma.

Connected sub-graphs identified by AAA are presented graphically to the user, e.g. in Fig. 3 for the new lemma *flinta* ('shotgun', 'flint-lock gun') one lexical unit is suggested and presented as a graph with the highest fit for the synset including *strzelba* 'shotgun'. On the screen, the purple oval represents the new lemma, synsets of the colours from yellow to red are elements of the identified sub-graph and green synsets do not belong to the suggestion but have been unfolded manually by the user. Synset colour expresses the strength of semantic fit between it and the new lemma: light yellow means weak strength of fit, darker yellow and red represents increasing strength. In the case of synsets presented as rectangles the fit was calculated on the basis of the Measure of Semantic Relatedness only ("weak fit" in [13], [18]), while octagons signal stronger fit supported by data from several knowledge sources ("strong fit" in [13], [18]).

The main change in the WordnetWeaver user interface is its full integration with WordnetLoom. Thus, suggestions are shown as hypernymic sub-graphs, but the user can browse relations of any type and add a new lexical unit for the new lemma with the help of any relation. In the previous version of WordnetWeaver [13] only hypernymic links could be browsed. A new lemma could be added to an existing synset member or as a new hypo/hypernym, but any other relation could not be directly used. WordnetWeaver was initially focused on expanding a wordnet along synonymy and hypernymy dimensions. That is why its effective use was limited to nouns, and the first attempts to apply it to verbs failed, as the verbal hypernymy structure is very limited. However knowledge sources assign

high values of semantic association to lemma pairs that are linked not only by synonymy/hypernymy, but also by other lexico-semantic relations, e.g. meronymy/holonymy. Thus new LUs suggested for a lemma can be also motivated by relations other than synonymy/hypernymy.

In contrast to plWordnetApp, the user interface of new version of WordnetWeaver, presented here, has been changed to the new one developed for WordnetLoom. As a results, see e.g. Fig. 3, the user now has access to any relation defined in the wordnet and new lexical units can be attached by any relation. A new algorithm of the layout generation prevents overlapping presentation of synsets on the screen.

WordnetWeaver was intended to be a tool for a team of linguists working simultaneously, rather than for an individual linguist. Thus the work procedure is based on assigning disjoint sets of lemmas, called 'packages', to particular linguists. Packages are numbered and a package can be selected using the text edit called `Package number`. Instead of dividing lemmas into packages on the basis of their alphabetic order, lemmas are semantically grouped by the clustering algorithm implemented in CLUTO system [19] applied to the results of the Measure of Semantic Relatedness, cf [13]. We did not expect perfect clustering, e.g. in terms of cluster purity, we only wanted to achieve a practical effect of groups generally semantically coherent. Created lemma groups appeared to be quite coherent, on average 2–3 semantic domains (e.g. names of professions, food, plants, minerals, etc.) can be noticed. During two years of intensive practical use this method based on the off-the-shelf clustering tool proved its value. In the new version of the WordnetWeaver we added a mechanism of temporal blocking a package for the linguist editing it. The mechanism is transparent to the users and blocking starts with the first editing actions and is automatically removed after the linguist has closed the client application or switched to editing of another package.

Quality of suggestions generated by the AAA depends on the local quality of the wordnet structure, e.g. if for there are no appropriate hypernyms for the new lemmas in the wordnet structure, the generated suggestion can be accidental. Thus, WordnetWeaver is equipped with mechanism of re-computing suggestions, which is activated on the user request (the button `Re-compute`). This facility can be especially helpful in case of constructing a new hypernymy sub-graph for some domain which was scarcely populated at the beginning. In the older version of WordnetWeaver the re-computation was run for all packages, in the present version the process is limited to one package only and can serve simultaneously several requests putting them into an internal queue.

V. Management Tools

The linguistic team is organised into two subgroups: wordnet editors and coordinators. The task of coordinators is to take care of the consistency of decisions made by different editors and to check the quality of changes introduced. In order to support the work of coordinators and facilitate the team cooperation a set of tools was developed. The tools are accessible via web pages in order to make usable even in the case of low bandwidth network connections.

First, browsing all changes is possible, e.g. adding a new wordnet element (like lexical unit, synset, relation link, etc.), deleting and modifying. Each change is registered together with information concerning the exact time and person who made it. An example screenshot is presented in Fig. 4. The filtering functionality – the top-right part of the screen – enables limiting the data to particular period, linguist or domain. Moreover, each team member can search for the source of the observed wordnet element, e.g. in order to get explanations concerning the reason of the introduction of some specific lexical unit (word sense).

On the additional web page, coordinators can observe the pace of work in relation to particular linguists, and on the basis of the collected data they are able also to plan the distribution of work among linguists.

Some diagnostic reports have been introduced, as well. Coordinators can obtain data concerning synsets without hypernyms and/or any other relation link[12]. These reports are the basis for correcting the structure and joining separated hypernymic sub-graphs into larger structures in those cases where it is supported by the linguistic data. The whole hypernymy graph is being constructed in plWordNet in the bottom-up direction, i.e. starting with more specific lexical units and their relations.

The set of management tools is being continuously extended but even its present version is frequently used by the coordinators and the linguistic team.

VI. Further Research

The vast majority of WordnetLoom is now implemented. Its new version described in this paper has been used for the last three months and about 9000 LUs and almost 20000 relation links have been added into plWordNet with its help. The idea of the primarily graph-based user interface in a wordnet editing tool seems to work well in practice, especially in the case of group work on expanding larger wordnet in the process driven by data extracted from large corpora. In such a process different parts of the wordnet are being developed simultaneously, and the linguists must follow the changes in an appropriate way.

Concerning the application development, we concentrate now on fixing problems which appeared during the work of linguists, as well, as minor improvements introduced in response to requests of linguists.

We work now on a significant extension of the AAA algorithm which is the basis for WordnetWeaver. We aim for the use of all types of relation links as a basis for expansion (now, only the hypernymy structure is used) and generating suggestions targeted at particular relations as description of the

[12]The presence of a synset which is not linked to any hypernym does not necessarily mean an error in the wordnet. In plWordNet every relation link must be supported by the linguistic analysis. Thus it is only required that there must be at least one relation link for a synset, and each lexical unit (but not lemma) must belong to exactly one synset.

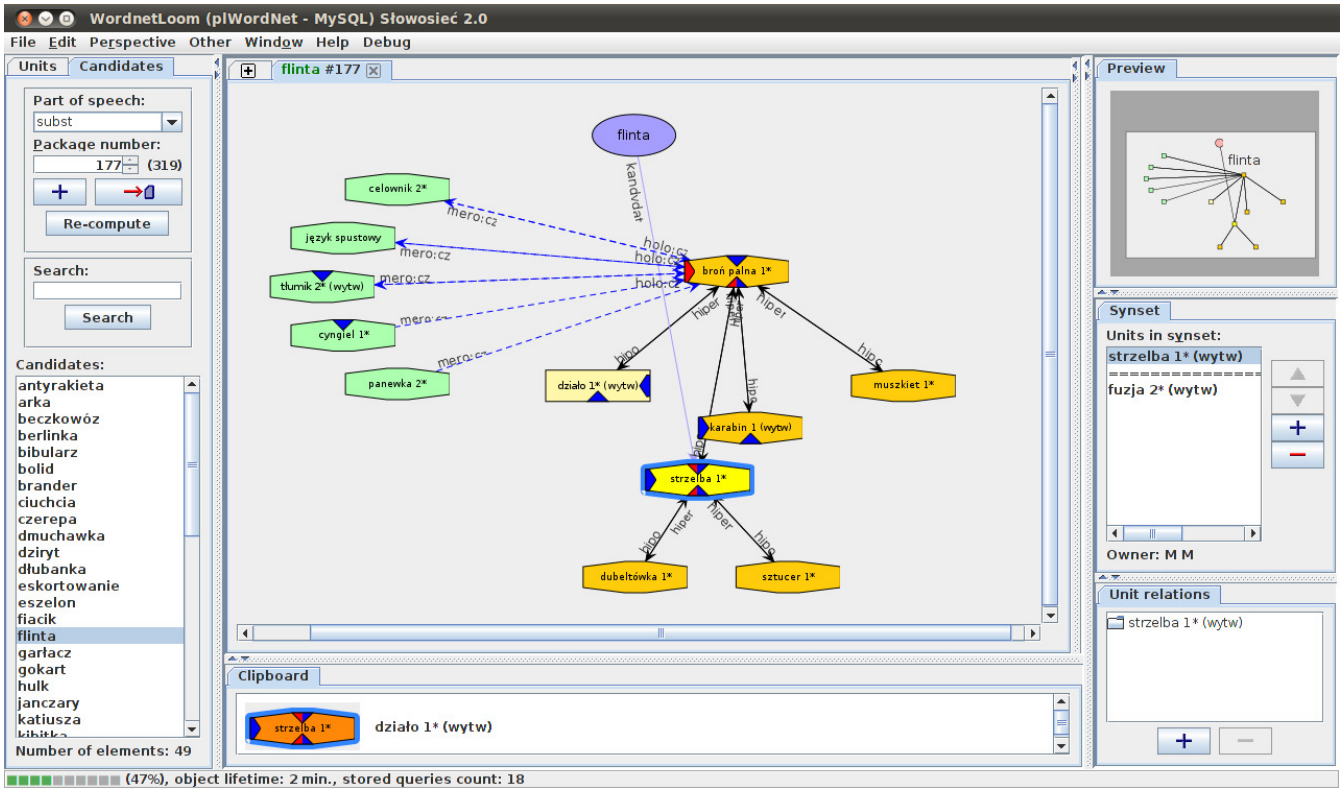


Fig. 3. Screenshot of Wordnet Weaver tool supporting semi-automated wordnet expansion (Glosses for synsets: *flinta* 'shotgun', *celownik* 'sight', *cyngiel* 'trigger', *język spustowy* 'trigger', *panewka* 'pan', *tłumik* 'silencer', *broń palna* 'firearm', *działo* 'gun', *karabin* 'rifle', *muszkiet* 'musket'), *strzelba* 'shotgun', *dubeltówka* 'double-barrelled gun', *sztucer* 'rifle')

attachment place, i.e. a synset selected as a point of attachment will be described by the type of relation by which a new lexical unit should be linked to it.

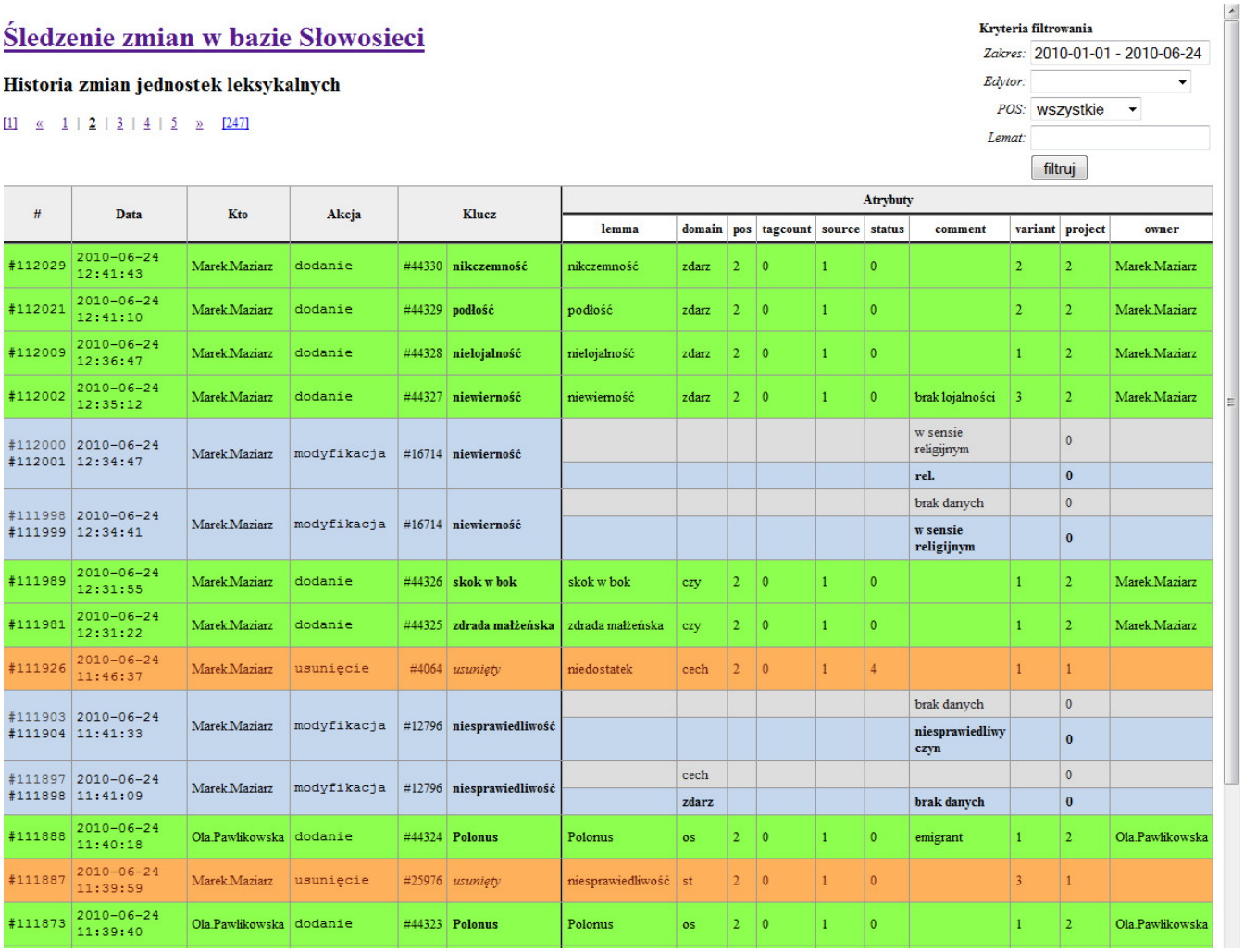
WordnetLoom can be used for editing other wordnets than plWordNet. The application uses UTF encoding for characters, relation types are defined in the XML-based format of wordnet data and the list of domains can be easily extended (now we use the Princeton WordNet list of domains). Lists of synset relations that are presented in vertical and horizontal directions are defined in the set-up file. Import and export to the Princeton WordNet file format has been implemented (but the use of WordnetWeaver requires preparation of the appropriate data knowledge sources). The suggestion presentation mechanism can work on the basis of any list of triples: word, synset, fit value. However, the AAA algorithm needs tuning to knowledge sources other than those used by us. WordnetLoom can be used for editing any network of lexical semantic relations or even an ontology. However, in the latter case, the limited, fixed set of the LU properties can be a strong limitation: concepts in an ontology are mostly described by an expandable attribute-value structures while lexical units are attached a limited, fixed set of properties in a wordnet.

ACKNOWLEDGMENT

Work financed by Innovative Economy Programme project POIG.01.01.02-14-013/09.

REFERENCES

[1] C. Fellbaum, Ed., *WordNet — An Electronic Lexical Database*. The MIT Press, 1998.
[2] G. A. Miller and C. Fellbaum, "Wordnet then and now," *Lang Resources & Evaluation*, vol. 41, pp. 209–214, 2007.
[3] R. I. Tengi, *Design and Implementation of the WordNet Lexical Database and Searching Software*. The MIT Press, 1998, ch. 4, pp. 105–127.
[4] P. Vossen, "EuroWordNet general document version 3," University of Amsterdam, Tech. Rep., 2002.
[5] A. Horák and P. Smrž, "New features of wordnet editor VisDic," *Romanian Journal of Information Science and Technology*, vol. 7, no. 1–2, pp. 201–213, 2004.
[6] T. Erjavec and D. Fišer, "Building the Slovene Wordnet: first steps, first problems," in *Proceedings of the Third International WordNet Conference — GWC 2006*, 2006.
[7] A. Horák, K. Pala, A. Rambousek, and M. Povolný, "DEBVisDic — first version of new client-server wordnet browsing and editing tool," in *Proceedings of the Third International WordNet Conference — GWC 2006*. Masaryk University, 2006, pp. 325–328. [Online]. Available: http://nlp.fi.muni.cz/publications/gwc2006_hales_pala_etal/gwc2006_hales_pala_etal.pdf
[8] V. Henrich and E. Hinrichs, "Gernedit - the germanet editing tool," in *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odjik, S. Piperidis, M. Rosnera, and D. Tapias, Eds. Valletta, Malta: European Language Resources Association (ELRA), may 2010.
[9] J. Kamps and M. Marx, "Visualizing WordNet structure," in *Proceedings of the 1st International Conference on Global WordNet,*, Mysore, India, 2002, pp. 182–186.
[10] C. Collins, "WordNet Explorer: Applying visualization principles to lexical semantics," Computational Linguistics Group, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada,



Śledzenie zmian w bazie Słowosieci

Historia zmian jednostek leksykalnych

[1] « 1 | **2** | 3 | 4 | 5 » [247]

Kryteria filtrowania

Zakres: 2010-01-01 - 2010-06-24
Edytor:
POS: wszystkie
Lemat:
filtruj

#	Data	Kto	Akcja	Klucz		Atrybuty									
						lemma	domain	pos	tagcount	source	status	comment	variant	project	owner
#112029	2010-06-24 12:41:43	Marek.Maziarz	dodanie	#44330	**nikczemność**	nikczemność	zdarz	2	0	1	0		2	2	Marek.Maziarz
#112021	2010-06-24 12:41:10	Marek.Maziarz	dodanie	#44329	**podłość**	podłość	zdarz	2	0	1	0		2	2	Marek.Maziarz
#112009	2010-06-24 12:36:47	Marek.Maziarz	dodanie	#44328	**nielojalność**	nielojalność	zdarz	2	0	1	0		1	2	Marek.Maziarz
#112002	2010-06-24 12:35:12	Marek.Maziarz	dodanie	#44327	**niewierność**	niewierność	zdarz	2	0	1	0	brak lojalności	3	2	Marek.Maziarz
#112000 #112001	2010-06-24 12:34:47	Marek.Maziarz	modyfikacja	#16714	**niewierność**							w sensie religijnym		0	
												rel.		**0**	
#111998 #111999	2010-06-24 12:34:41	Marek.Maziarz	modyfikacja	#16714	**niewierność**							brak danych		0	
												w sensie religijnym		**0**	
#111989	2010-06-24 12:31:55	Marek.Maziarz	dodanie	#44326	**skok w bok**	skok w bok	czy	2	0	1	0		1	2	Marek.Maziarz
#111981	2010-06-24 12:31:22	Marek.Maziarz	dodanie	#44325	**zdrada małżeńska**	zdrada małżeńska	czy	2	0	1	0		1	2	Marek.Maziarz
#111926	2010-06-24 11:46:37	Marek.Maziarz	usunięcie	#4064	*usunięty*	niedostatek	cech	2	0	1	4		1	1	
#111903 #111904	2010-06-24 11:41:33	Marek.Maziarz	modyfikacja	#12796	**niesprawiedliwość**							brak danych		0	
												niesprawiedliwy czyn		**0**	
#111897 #111898	2010-06-24 11:41:09	Marek.Maziarz	modyfikacja	#12796	**niesprawiedliwość**		cech							0	
							zdarz					**brak danych**		**0**	
#111888	2010-06-24 11:40:18	Ola.Pawlikowska	dodanie	#44324	**Polonus**	Polonus	os	2	0	1	0	emigrant	1	2	Ola.Pawlikowska
#111887	2010-06-24 11:39:59	Marek.Maziarz	usunięcie	#25976	*usunięty*	niesprawiedliwość	st	2	0	1	0		3	1	
#111873	2010-06-24 11:39:40	Ola.Pawlikowska	dodanie	#44323	**Polonus**	Polonus	os	2	0	1	0		1	2	Ola.Pawlikowska

Fig. 4. Screenshot of the WWW-based browser of changes introduced in plWordNet.

Technical Report, 2008. [Online]. Available: http://faculty.uoit.ca/collins/publications/docs/TR-wordnet-explorer.pdf

[11] J. Szymański, "Wordventure – cooperative wordnet editor. Architecture for lexical semantic aquisition," in *Proceedings of the international conference on Knowledge Engineering and ontology developement, INSTICC*, Funchal-Madeira, Portugal, 2009.

[12] M. Derwojedowa, M. Piasecki, S. Szpakowicz, M. Zawisławska, and B. Broda, "Words, concepts and relations in the construction of Polish WordNet," in *Proceedings of the Global WordNet Conference, Seged, Hungary January 22–25 2008*, A. Tanâcs, D. Csendes, V. Vincze, C. Fellbaum, and P. Vossen, Eds. University of Szeged, 2008, pp. 162–177. [Online]. Available: http://www.plwordnet.pwr.wroc.pl/main/content/files/publications/gwc08-plWN-paper.pdf

[13] M. Piasecki, S. Szpakowicz, and B. Broda, *A Wordnet from the Ground Up*. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 2009. [Online]. Available: http://www.plwordnet.pwr.wroc.pl/main/content/files/publications/A_Wordnet_from_the_Ground_Up.pdf

[14] M. Derwojedowa, M. Głabska, M. Piasecki, J. Rabiega-Wiśniewska, S. Szpakowicz, and M. Zawisławska, "plWordNet 1.0 — The Polish Wordnet," April 2009, online access to the database of plWordNet 1.0: www.plwordnet.pwr.wroc.pl. [Online]. Available: http://plwordnet.pwr.wroc.pl/browser/?lang=en

[15] B. Hamp and H. Feldweg, "GermaNet — a lexical-semantic net for German," in *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications, Madrid*. ACL, 1997.

[16] M. Derwojedowa, M. Piasecki, S. Szpakowicz, and M. Zawisławska, "Polish WordNet on a shoestring," in *Proceedings of Biannual Conference of the Society for Computational Linguistics and Language Technology, Tübingen, April 11-13 2007*. Universität Tübingen, 2007, pp. 169–178. [Online]. Available: http://www.plwordnet.pwr.wroc.pl/main/content/files/publications/piasecki-gldv07-final.pdf

[17] M. Woliński, "Morfeusz – a practical tool for the morphological analysis of Polish." ser. Advances in Soft Computing, M. A. Kłopotek, S. T. Wierzchoń, and K. Trojanowski, Eds. Berlin: Springer, 2006, pp. 511–520.

[18] M. Piasecki, B. Broda, M. Głąbska, M. Marcińczuk, and S. Szpakowicz, "Semi-automatic expansion of polish wordnet based on activation-area attachment," in *Recent Advances in Intelligent Information Systems*. Academic Publishing House EXIT, 2009, pp. 247–260. [Online]. Available: http://iis.ipipan.waw.pl/2009/proceedings/iis09-25.pdf

[19] G. Karypis, "CLUTO - a clustering toolkit," Tech. Rep. #02-017, nov 2003.

[20] A. Zgrzywa, Ed., *Proceedings of Multimedia and Network Information Systems*. Oficyna Wydawnicza Politechniki Wrocławskiej, 2006.

[21] M. A. Kłopotek, S. T. Wierzchoń, and K. Trojanowski, Eds., *Intelligent Information Processing and Web Mining – Proceedings of the International IIS: IIPWM'06 Conference held in Wisła, Poland, June 2006*, ser. Advances in Soft Computing. Berlin: Springer, 2006.