Experiences of Lexicographers and Computer Scientists in Validating Estonian Wordnet with Test Patterns

Ahti Lohk[1], Heili Orav[2], Kadri Vare[2] and **Leo Võhandu[1]**

[1]Department of Informatics, Tallinn University of Technology, Tallinn, Estonia
[2]Department of Computer Science, University of Tartu, Tartu, Estonia

<{ahti.lohk, leo.vohandu}@ttu.ee, {heili.orav, kadri.vare}@ut.ee>

Abstract

New concepts and semantic relations are constantly added to Estonian Wordnet (EstWN) to increase its size. In addition to this, with the use of test patterns, the validation of EstWN hierarchies is also performed. This parallel work was carried out over the past four years (2011-2014) with 10 different EstWN versions (60-70). This has been a collaboration between the creators of test patterns and the lexicographers currently working on EstWN. This paper describes the usage of test patterns from the points of views of information scientists (the creators of test patterns) as well as the users (lexicographers). Using EstWN as an example, we illustrate how the continuous use of test patterns has led to significant improvement of the semantic hierarchies in EstWN.

1 Introduction and background

1.1 About Estonian Wordnet

The Estonian Wordnet began as a part of the EuroWordNet project (Vossen, 1998) and was built by translating basic concepts from English to allow for the monolingual extension. Words (literals) to be included were selected on a frequency basis from corpora. Extensions have been compiled manually from Estonian monolingual dictionaries and other monolingual resources. In this process, several methods have been used. For example, domain-specific methods, i.e. semantic fields like architecture, transportation, etc. have been covered. Moreover, there have been endeavors to automatically add derivatives and the results have been used in the sense disambiguation process. Version 70 of EstWN consists of 67,674 *synsets*, including 110,869 *lexical units*.

1.2 Previous experience of validation

Before the introduction of test patterns, the EstWN was validated and revised by adding new synsets and semantic relations into its semantic network. Information about new lexical concepts (synsets) originated from the Estonian language explanatory dictionary (EKSS[1]), text corpora and even from feedback on applying EstWN to the word sense disambiguation (WSD) task (Kahusk and Vider, 2002). In addition, EstWN participated in the META-NORD project, which aims to link and validate Nordic and Baltic wordnets (*Danish, Estonian, Finnish, Icelandic, Latvian, Lithuanian, Norwegian and Swedish*) and make these resources widely available for different categories of user communities in academia and in the industry. Under this project, the preliminary task is to "*upgrade several wordnet resources to agreed standards" "and let them undergo cross-lingual comparison and validation in order to ensure that they become of the highest possible quality and usefulness*" (Pedersen et al., 2012).

The first attempt to check the structure of EstWN took place with version 55 (by the first author of this paper). One of the aspects studied was the number of branches a *synset* goes through before arriving at one or several *root synsets*. These results were presented at the Estonian Applied Linguistics Conference in spring 2011, where Kadri Vider[2] provided our first feedback. Her comments elucidated that EstWN requires this kind of structure checking. In the same year, the first attempt was made to validate EstWN with the test pattern[3] of *closed subset*. Test pattern instances were evaluated by Kadri Vare and some of the results were reflected in two papers (Lohk et al., 2012a), (Lohk et al., 2012b). Later Lohk et

[1] http://www.eki.ee/dict/ekss/

[2] A computational linguist from the University of Tartu.

[3] Test pattern is a description of a substructure with a specific nature in the wordnet semantic network (intended to validate the semantic hierarchies of wordnet).

al. (2014b) discovered more test patterns, all related to multiple inheritance cases. Presently, there is a system of ten test patterns (Lohk, 2015).

This paper aims to introduce these test patterns and prove that the usage of the test patterns to validate semantic hierarchies of wordnet may significantly improve the wordnet structure. In addition, lexicographers Heili Orav and Kadri Vare share their experiences of working with these test pattern instances (Section 5).

The paper is structured as follows: Section 2 elaborates on the motivation for this work. Section 3 provides a general description of the test patterns, followed by examples of test pattern instances. Section 4 proves the efficiency of test pattern instances in validating the semantic hierarchies of wordnet. Section 5 describes the experiences of lexicographers in using test pattern instances.

2 Motivation

There are many reasons for why test patterns should be chosen as a way to validate *multiple inheritance* in the wordnet hierarchical structure (formed by its semantics). To begin with, due to the nature of *multiple inheritance*, it requires checking. More precisely, multiple inheritance is prone to semantic errors:

1) Inappropriate use of multiple inheritance (Kaplan and Schubert, 2001). There are many cases where multiple inheritance is not used as a conjunction of two properties (Gangemi et al., 2001).
2) Sometimes an IS-A relation is used instead of other semantic relations (Martin, 2003). Multiple inheritance makes it possible to compare relations that connect the parents of a synset.
3) In many cases, multiple inheritance causes topological rings (Liu et al., 2004), (Richens, 2008). According to (Liu et al., 2004), one synset cannot inherit properties from both parents.
4) Multiple inheritance may refer to a short cut problem (Fischer, 1997), (Liu et al., 2004), (Richens, 2008). One synset has a two-fold connection to another one, both directly and indirectly. The direct link is illegal.
5) Multiple inheritance may refer to dangling uplinks in the hierarchical structure (Šmrz, 2004).

Secondly, the use of test patterns has many advantages:

1) Using a test is always quicker than "*[doing] a full revision in top-down or alphabetical order*" (Čapek, 2012).
2) Use of "*manual verification and correction*" is the most reliable. (Lindén and Niemi, 2014).
3) Test pattern instances highlight substructures that refer to possible errors and they simplify the work of the expert lexicographer (Lohk et al., 2012a), (Lohk et al., 2012b), (Lohk et al., 2014b).
4) Test patterns are applicable to wordnets in any language (Lohk et al., 2014c).

3 Test patterns

3.1 General knowledge about test patterns

As mentioned above, test patterns, by their nature, are descriptions of substructures with a specific nature in the wordnet semantic hierarchy that are intended to validate its structure. All patterns have the property of *multiple inheritance*. In most cases, there is a lexical polysemy behind *multiple inheritance*. In the remaining cases, there are *synsets* that simultaneously inherit specific and general concepts (test pattern of *short cut*).

Test pattern instances help to detect possible errors in the semantic hierarchies of wordnet. Each test pattern provides a different perspective to the semantic hierarchy. Thus, they vary in their capability to discover various types of possible semantic errors. Test pattern instances are identified by programs and have to be validated by an expert lexicographer.

Test pattern structures partially or entirely overlap with each other. However, they have different perspectives to the substructures of hierarchies and may typically point to different semantic errors therein.

There are only two ways to cover all *multiple inheritance* cases in the certain semantic hierarchy of a wordnet – by using test pattern instances of *closed subset* or test pattern instances of *ring* and *synset with many roots* together.

We developed algorithms and created programs (in the framework of the doctoral thesis of (Lohk, 2015)) to automatically find instances of the different types of test patterns. However, some algorithms and programs are implemented to semi-automatically find instances of different types of test patterns. **Table 1** gives an overview of the developed test patterns and information about the automation level of finding their instances. This table illustrates that six of the test patterns are implemented to find their instances in an automatic way and the remaining four in a semi-automatic way. In addition, it should be mentioned that the first two patterns (*short cut* and *ring*) are inspired by other authors (Fischer, 1997), (Liu et al., 2004), (Richens, 2008). Test patterns with a gray background are all the *closed subset* patterns, however, the second and third ones have a specific property. Moreover, the test pattern instances of *synset with many roots* may in some cases correspond to the substructure called *dangling uplink* noted by (Koeva et al., 2004) and (Šmrz, 2004).

Test pattern	**Automation level**
Short cut	*automatic*
Ring	*automatic*
Closed subset	semi-automatic
Closed subset with a root	semi-automatic
The largest closed subset	semi-automatic
Dense component	automatic
Heart-shaped substructure	automatic
Synset with many roots	automatic
“Compound” pattern	automatic
Connected roots	semi-automatic

Table 1: *Automation level of finding test pattern instances*

Even though there exist ten test patterns (Table 1), only the instances findable in an automatic way were delivered to the lexicographer.

Below, four of them are described, while *short cut* and *ring* are considered by their authors and the main author of this paper. However, it may be useful to mention that *short cut* indicates redundancy in the semantic hierarchy and *ring* may refer to problematic synsets, which are simultaneously co-hyponyms and co-hypernyms and additions from the same domain category (Liu et al., 2004).

All of the following examples are described by the first author of this paper. Moreover, all ten test patterns are described as mathematical models (more precisely, as graphs) in the thesis of (Lohk, 2015).

In the examples, every synset is equipped with the equivalent synonyms from Princeton WordNet Version 1.5 and begins with an abbreviation “(Eq_s)”. If the equivalent synonyms are unknown, free translation has been used.

3.2 Dense component

The *dense component* pattern provides an opportunity to uncover substructures where, due to *multiple inheritance*, the density of the interrelated concepts in the semantic hierarchy is higher (Lohk et al., 2014a), (Lohk et al., 2014b). This pattern contains at least two ambiguous concepts (as in Figure 1 {hotel_1} and “hostel”), which have a minimum of two identical parents (“a housing enterprise” and “accommodation building”). The benefit of this pattern is its ability to uncover all *regular polysemy* cases that reveal themselves as the *regularity of multiple inheritance*.

The lexicographer has to establish:

- whether that kind of regularity is justified, and
- whether *multiple inheritance* can be extended to another *synset*(s)

In order to better understand the semantic field of the dense component in Figure 1, the *synsets* with dotted lines are additional information to the *dense component* (*synsets* with bold lines) to grasp its content more clearly. The first number after in the brackets the *synset* indicates the number of subordinates inside the *dense component*. The second number in the brackets displays the count of all the subordinates for that *synset*.

It is a well-known fact that there are several concepts related to polysemic patterns (Langemets, 2010). Based on Figure 1, {hotel_1} and “hostel” describe that kind of pattern through *institution-building*. Checking the concept(s) additional to {hotel_1} and “hostel”, {motel_1, …} is found which in its nature is quite similar to {hotel_1} and “hostel”. Hence, it appears reasonable to also connect it to “accommodation building”.

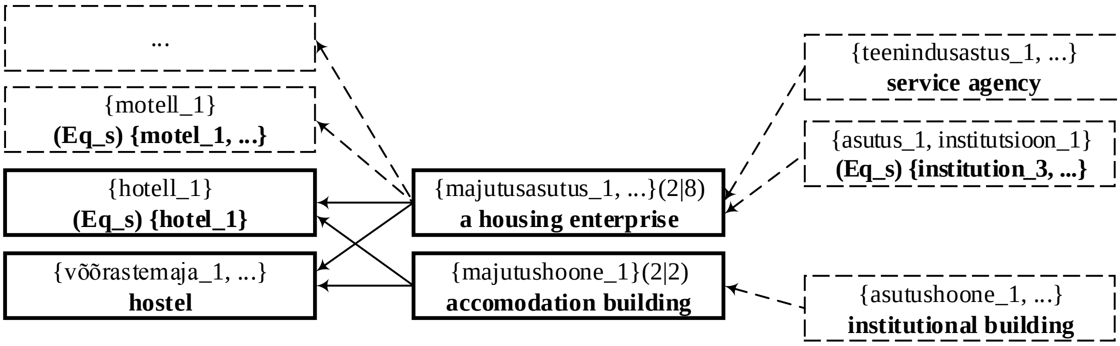

Figure 1*. An instance of the dense component (rotated 90 degrees)*

In the latest version of EstWN, it emerged that {hotel_1} and "hostel" are no longer connected to building through a *hypernymy* relation. (Instead, the connection is through *near_synonymy*.) Meanwhile, in the current version of Princeton WordNet[4], {hotel_1} is only a building and {hostel_1} is its subordinate. For a solution, let us look at another concept similar to motel, hotel, and hostel – the hospital. EstWN organizes this concept into two *synsets*. Firstly, it denotes a *medical institution*, and secondly, a *medical building*. A similar idea is followed in Princeton WordNet. Thus, in both wordnets, *hospital* is related to an *institution* as well as a *building*. According to this example, it is advised to organize the concepts *hotel*, *motel* and *hostel* in a similar manner.

3.3 Heart-shaped substructure

The *heart-shaped substructure* pattern describes the substructure in the wordnet hierarchy where two synsets (in Figure 2, {homoepathy_1} and "mud cure, mud treatment") along with their two parents are interconnected due to a common parent ({curative_1, cure_1}) as well as through a hypernymy relationship between another one of their parents ({naturopathy_1} and {alternative medicine_1, …}).

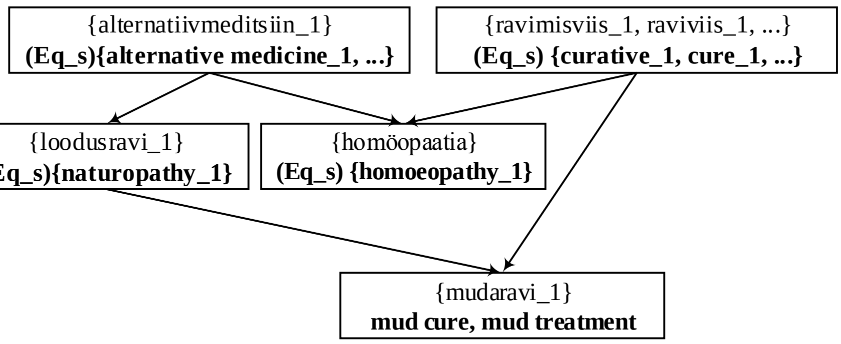

Figure 2*. An instance of the heart-shaped substructure*

In the report file on the instances of a *heart-shaped substructure* delivered to lexicographers, additional subordinates of the two topmost nodes are shown. This helps to assess why these two *synsets* with two parents are so specific that they join superordinates but their co-members under both parents are not linked.

Secondly, this pattern indicates an instance, where a super-concept ({curative_1, cure_1, …}) seems to be connected to a sub-concept from a different taxonomy level ("mud cure, mud treatment"). On the one hand, this situation might be a particular feature of the language, but on the other hand, it might refer to an error.

An example of a *heart-shaped substructure* in Figure 2 originates from (Lohk et al., 2014b). The question arises why {homoeopathy_1} is not a subcase of {naturopathy_1}. Secondly, are "mud cure, mud treatment" and {homoeopathy_1} subcases of {alternative medicine_1} or of {curative_1, cure_1, …}? On the basis of the definitions of these concepts, the lexicographers decided that both are subcases of {curative_1, cure_1, …} and that {alternative medicine_1} is connected to them via a holonymy relation.

There is still no thorough analysis of the *heart-shaped substructure*. Therefore, there is no such

[4] http://wordnetweb.princeton.edu/perl/webwn

instance in the latest version of EstWN. In addition, as discovered in (Lohk and Võhandu, 2014), most of the cases of *heart-shaped substructure*s in Princeton WordNet pointed to situations where instead of a *hypernymy* relation there should have been a *role* or *type* relation.

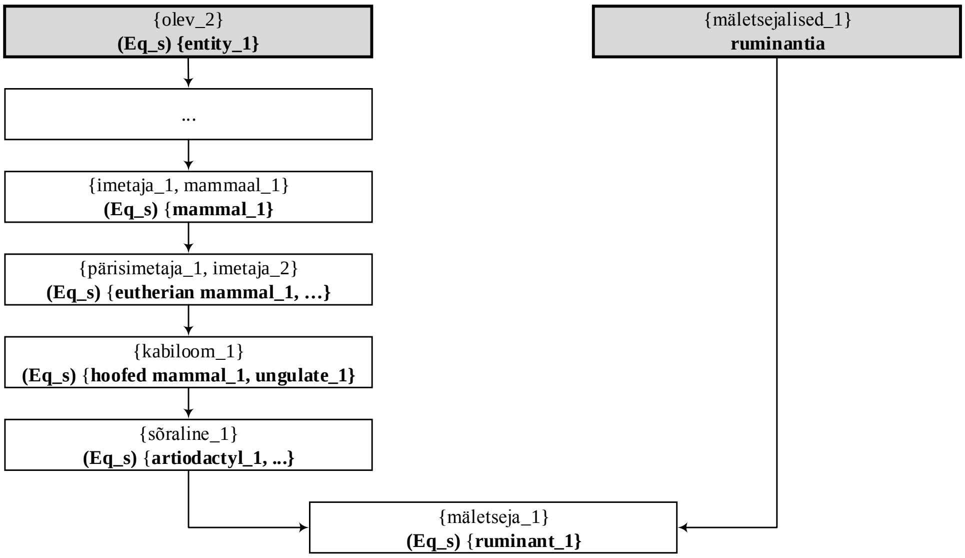

Figure 3. *An instance of the connected roots*

3.4 Synsets with many roots

Quite a similar pattern to rings is the *synset* with many roots. This pattern differs from the former one by its unconnected branches. On the one hand, this signifies that some of the detectable errors are similar to rings. On the other hand, this pattern is capable of discovering errors related to *root synsets*. Figure 3 demonstrates how one *root synset* is *a dangling uplink*[5] – "ruminant animals". It means that the *synset* ({ruminant_1}) is connected to the second parent ("ruminantia") which represents a *root synset*, but in fact, is carrying the ower-level concept. The *root synset* "ruminantia" is a taxon, i.e. it represents a group of animals with particular properties. Therefore, it was correct to change the *hypernymy* relationship between {ruminant_1} and „ruminantia" to holonymy. Thus, {ruminant_1} belongs to the group "ruminantia".

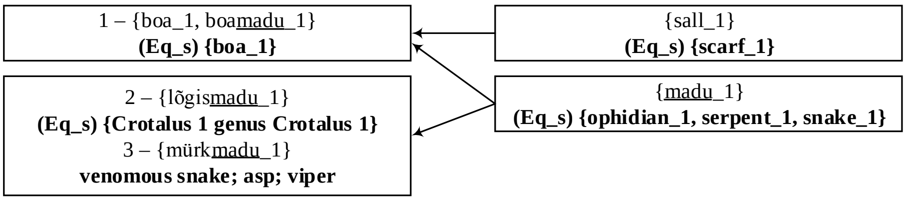

Figure 4. *An instance of the "compound" pattern*

3.5 Substructure that considers the content of synsets ("compound" pattern)

(Nadig et al., 2008) consider a relationship between *synsets* where a member of a *synset* is a suffix to the member of another *synset*. They utilize examples such as {work}, {paperwork}, and {racing}, {auto racing, car racing}. In that manner, it is possible to check whether that *synset* has a *hypernymy* relation. In this pattern, the idea of (Nadig et al., 2008) is employed to uncover all the cases where this condition is true. Additionally, we have to consider that at least one of the subordinates has an additional superordinate as in Figure 4, where {boa_1} has a superordinate {scarf_1}. In that case, the lexicographer must consider why {boa_1} with an extra superordinate did not have any connections to the other subordinates. Upon checking this additional concept ({scarf_1}), it emerges that it is totally unsuitable because while {boa_1} is a *serpent*, *scarf* is a *garment*. However, *scarf* is still related to *boa*, but in a different meaning {boa_2, feather boa_1}.

[5] *Dangling uplink* is a special case of the *synset with many roots*.

4 Experiences of lexicographers in using test pattern instances

The activities of a lexicographer are rather diverse. Compiling a thesaurus requires access to vast amounts of linguistic data (e.g. corpora, different dictionaries, databases) as well as knowledge of how to analyze these data.

Test patterns provide lexicographers with a broader overview than daily work with a lexicographic tool could ever give. All the patterns were checked individually. In many cases, additional descriptions of usage context or definitions help to ascertain the correct relations between the concepts and may also provide additional relations found to be missing.

On occasion, synsets with many hypernyms were left unaltered. For example, *morphine* is simultaneously both a narcotic and pain medicine. This illustrates a well-known problem: *"Rigidity property plays an important role when we distinguish semantic relations of* **type** *and* **role***"* because "*every type is a rigid concept and every role is a non-rigid concept*" (Hicks and Herold, 2011). It is suspected that the *hyponymy* relation may sometimes be a *role* or *type* relationship.

There were also instances where a hypernym had several hyponyms which in turn indicated a problem, namely that some hyponyms had hypernyms that were too general. Revising the hypernymy trees often reduced the amount of direct hyponyms, resulting in a more precise and systematic hierarchy.

Thus, lexicographer should also know how to use their own intuition in the decision-making process. As these test patterns only indicate possible problems, it is not sensible to apply test patterns automatically. However, it could be very useful, if the test pattern results ran simultaneously in a wordnet editing tool, so the lexicographer is provided with complementary information.

5 Iterative evolution of EstWN

Applying the test patterns to EstWN has taken place gradually. As mentioned earlier, we began validating EstWN with the *closed subset* test pattern. At that time, we studied approximately 20 instances of EstWN and Princeton WordNet. Some of the results are reflected in two joint papers with Kadri Vare (Lohk et al., 2012a) and (Lohk et al., 2012b). Later, we started to use *short cut* as well as other patterns.

In the iterative evolution of EstWN, test pattern instances were separated with help of our programs and subsequently delivered to lexicographers who validated all instances and corrected wordnet semantic hierarchies where necessary.

Table 2 reflects the number of test pattern instances over 11 EstWN versions. As background information, the noun roots, verb roots and multiple inheritance cases are also presented. Every number in this table indicates the condition of a specific version in the light of the number of test pattern instances. These numbers are found immediately after the addition of new concepts and semantic relations, and the release of the new version. Thus, the correction of semantic hierarchies is revealed in the next version of wordnet.

The bold font in Table 2 indicates the versions in which a specific pattern was applied. We may notice that in the range from 60 to 62 no test patterns are used. As a matter of fact, at that time we conducted some experiments with the *closed subset* pattern for our first two papers. Beside the numbers of test pattern instances, it is important to observe the number of multiple inheritance cases, as every test pattern instance contains at least one. The last row in this table confirms that one multiple inheritance case may be contained in many different types of test pattern instances, while the total of the last row of instances (7+21+30+0+3+7) is bigger than the multiple inheritance cases (51).

The largest changes in the number of multiple inheritance cases appear when *dense components* are taken into use in version 66. This is due to the fact that dense component contains at least two or more multiple inheritance cases in one instance. In the paper of (Lohk et al., 2014a), it was discovered that only 12% (14) of 121 dense component instances do not need any correction. Nevertheless, the next version (67) revealed 8 new instances.

The decrease in the number of multiple inheritance cases continues even after version 67 when two more patterns are applied (*heart-shaped substructure* and *"compound" pattern*). In the last version, there are only 3 *dense component* instances and 0 *heart-shaped substructure* instances. Comparing the numbers of multiple inheritance cases in versions 66 and 70, it is noted that the last number (51) is approximately 32 times smaller, i.e. multiple inheritance cases have been shrunk by approximately 97%.

Version	Noun roots	Verb roots	Multiple inheritance cases	Short cuts	Rings	Synset with many roots	Heart-shaped sub-structure	Dense component	“Compound“ pattern
60	142	24	1,296	235	3,445	1,123	1,825	104	301
61	183	22	1,592	259	3,560	1,309	1,861	121	380
62	102	16	1,700	299	3,777	1,084	1,941	128	415
63	114	16	1,815	**321**	3,831	1,137	2,103	141	447
64	149	15	1,893	**337**	3,882	1,173	2,232	149	471
65	248	14	1,717	**194**	2,171	791	**451**	132	459
66	144	4	1,677	**119**	1,796	613	259	**121**	671
67	129	4	1,164	**79**	928	477	**167**	**24**	**407**
68	131	4	691	**60**	**537**	232	**38**	**18**	**54**
69	121	4	102	**18**	**291**	35	**1**	**8**	23
70	118	4	51	**7**	**21**	**30**	**0**	**3**	**7**

***Table 2**: A numerical overview of EstWN spanning 11 versions*

6 Conclusion and future works

The main collaboration between computer scientists and lexicographers in order to validate EstWN (version 60) began with the *closed subset* test pattern. The *closed subset* was successful in finding possible errors in semantic relations. Later, nine other test patterns dealing with multiple inheritance were developed (see more: Lohk, 2015). Two patterns, namely *short cut* and *ring patterns* are inspired from different authors and one pattern can in certain cases include a *dangling uplink*. In this paper, six test patterns were described but the examples covered four test patterns.

Typically, the work for using test patterns was organized as follows: the first author of this paper generated the instances of test patterns, then based on that document, the lexicographer made corrections using the EstWN editing tool.

The experience of validating Estonian Wordnet assured that the continuous usage of test patterns can significantly improve the semantic hierarchy. Multiple inheritance decreased 32 times or 97% in the last five versions of EstWN.

In the future, we plan to apply these test patterns to other types of semantic relations, for instance to *near synonymy*, *fuzzynymy* and *holonymy*. Moreover, as there are about 70 wordnets in the world, we believe that applying these test patterns to them may “*automatically characterize their modelling decisions (i.e. potential modelling errors)”*[6].

Reference

Čapek, T., 2012. SENEQA-System for Quality Testing of Wordnet Data, in: Proceedings of the 6th International Global Wordnet Conference. Toyohashi University of Technology, Matsue, Japan, pp. 400–404.

Gangemi, A., Guarino, N., Oltramari, A., 2001. Conceptual Analysis of Lexical Taxonomies: The Case of WordNet Top-Level, in: Proceedings of the International Conference on Formal Ontology in Information Systems-Volume 2001. ACM, pp. 285–296.

Kaplan, A.N., Schubert, L.K., 2001. Measuring and Improving the Quality of World Knowledge Extracted from WordNet (No. 751). The University of Rochester Computer Science Department, Rochester, New York.

Koeva, S., Mihov, S., Tinchev, T., 2004. Bulgarian Wordnet–Structure and Validation. Romanian J. Inf. Sci. Technol. 7, 61–78.

Langemets, M., 2010. Nimisõna süstemaatiline polüseemia eesti keeles ja selle esitus eesti keelevaras. Eesti Keele Sihtasutus, Tallinn, Eesti.

[6] Comment by a reviewer.

Lindén, K., Niemi, J., 2014. Is It Possible to Create a Very Large Wordnet in 100 Days? An Evaluation. Language Resources and Evaluation 48, 191–201.

Liu, Y., Yu, J., Wen, Z., Yu, S., 2004. Two Kinds of Hypernymy Faults in WordNet: the Cases of Ring and Isolator, in: Proceedings of the 2nd Global Wordnet Conference. Brno, Czech Republic, pp. 347–351.

Lohk, A., 2015. A System of Test Patterns to Check and Validate the Semantic Hierarchies of Wordnet-type Dictionaries. Tallinn University of Technology, Tallinn, Estonia.

Lohk, A., Allik, K., Orav, H., Võhandu, L., 2014a. Dense Component in the Structure of Wordnet, in: Proceedings of the 9th International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), Reykjavik, Iceland, pp. 1134–1139.

Lohk, A., Norta, A., Orav, H., Võhandu, L., 2014b. New Test Patterns to Check the Hierarchical Structure of Wordnets, in: Information and Software Technologies. Springer, pp. 110–120.

Lohk, A., Orav, H., Võhandu, L., 2014c. Some Structural Tests for WordNet with Results. Proceedings of the 7th Global Wordnet Conference 313–317.

Lohk, A., Vare, K., Võhandu, L., 2012a. Visual Study of Estonian Wordnet Using Bipartite Graphs and Minimal Crossing Algorithm, in: Proceedings of the 6th International Global Wordnet Conference. Matsue, Japan, pp. 167–173.

Lohk, A., Vare, K., Võhandu, L., 2012b. First Steps in Checking and Comparing Princeton Wordnet and Estonian Wordnet, in: Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH. Association for Computational Linguistics (ACL), pp. 25–29.

Lohk, A., Võhandu, L., 2014. Independent Interactive Testing of Interactive Relational Systems, in: Gruca, D.A., Czachórski, T., Kozielski, S. (Eds.), Man-Machine Interactions 3, Advances in Intelligent Systems and Computing. Springer International Publishing, pp. 63–70.

Martin, P., 2003. Correction and Extension of WordNet 1.7, in: Conceptual Structures for Knowledge Creation and Communication. Springer, pp. 160–173.

Nadig, R., Ramanand, J., Bhattacharyya, P., 2008. Automatic Evaluation of WordNet Synonyms and Hypernyms, in: Proceedings of ICON-2008: 6th International Conference on Natural Language Processing. CDAC Pune, India.

Pedersen, B.S., Forsberg, M., Borin, L., Lindén, K., Orav, H., Rögnvaldsson, E., 2012. Linking and Validating Nordic and Baltic wordnets, in: Proceedings of the 6th International Global Wordnet Conference. Matsue, Japan, pp. 254–260.

Richens, T., 2008. Anomalies in the Wordnet Verb Hierarchy, in: Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1. Association for Computational Linguistics (ACL), pp. 729–736.

Šmrz, P., 2004. Quality Control and Checking for Wordnet Development: A Case Study of BalkaNet. Science and Technology 7, 173–181.

Vossen, P., 1998. Introduction to EuroWordNet. Computers and the Humanities 32, 73–89.