THESIS ON INFORMATICS AND SYSTEM ENGINEERING C105

A System of Test Patterns to Check and Validate the Semantic Hierarchies of Wordnet-type Dictionaries

AHTI LOHK

TUT
PRESS

TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology
Department of Informatics

This dissertation was accepted for the defense of the degree of Doctor of Philosophy in Computer and Systems Engineering on July 27, 2015.

Supervisor: Prof. Emer. Leo Võhandu
Department of Informatics
Tallinn University of Technology

Opponents: Prof. Christiane D. Fellbaum
Department of Computer Science
Princeton University

Prof. Emer. Mare Koit
Department of Computer Science
University of Tartu

Defense of the thesis: August 28, 2015

Declaration:
I hereby declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology, has not been submitted for any academic degree.

/Ahti Lohk/

ISSN 1406-4731
ISBN 978-9949-23-831-6 (publication)
ISBN 978-9949-23-832-3 (PDF)

INFORMAATIKA JA SÜSTEEMITEHNIKA C105

Testmustrite süsteem *wordnet*-tüüpi sõnastike semantiliste hierarhiate kontrollimiseks ja valideerimiseks

AHTI LOHK

TTÜ
KIRJASTUS

TABLE OF CONTENTS

LIST OF PUBLICATIONS

All publications are reprinted in the appendices of the thesis.

I. Lohk, A.; Vare, K.; Võhandu, L. (2012). First Steps in Checking and Comparing Princeton WordNet and Estonian Wordnet. In: Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH: EACL 2012; April 23 - 24 2012; Avignon France. 2012, pp. 25 - 29.

II. Lohk, A.; Tilk, O.; Võhandu, L. (2013). How to create order in large closed subsets of wordnet-type dictionaries. Eesti Rakenduslingvistika Ühingu aastaraamat, 9, pp. 149 - 160.

III. Lohk, Ahti; Allik, Kaarel; Orav, Heili; Võhandu, Leo (2014). Dense Components in the Structure of WordNet. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14): LREC2014, Reykjavik, Iceland, May 26-31, 2014. (Edit.) Nicoletta Calzolari and Khalid Choukri and Thierry Declerck and Hrafn Loftsson and Bente Maegaard an. ELRA, 2014, pp. 1135 - 1139.

IV. Lohk, A.; Võhandu, L. (2014). Independent Interactive Testing of Interactive Relational Systems. A. Gruca, T. Czachórski, S. Kozielski (Toim.). Man-Machine Interactions 3 (63 - 70). Springer

V. Lohk, A.; Orav, H.; Võhandu, L. (2014). Some structural tests for WordNet with results. H. Orav, C. Fellbaum, P. Vossen (Edit.). Proceedings of the Seventh Global Wordnet Conference (313 - 317). Tartu University Press

VI. Lohk, A; Norta, A; Orav, H; Võhandu, L. (2014). New Test Patterns to Check the Hierarchical Structure of Wordnets. G. Dregvaite, R. Damasevicius (Edit.). Information and Software Technologies: 20th International Conference, ICIST 2014, Druskininkai, Lithuania, October 9-10, 2014. Proceedings (110 - 120). Springer

Author`s Contribution to the Publications

I. The author's main contribution was implementing the modified equivalent classes' algorithm for certain substructure (instances of *closed subset* pattern) in the semantic hierarchies of wordnet and the minimal crossing algorithm for small *closed subsets* represented as bipartite graphs.

II. The author's contribution was everything except the description of very fast two-step method and its results on wordnets graphs.

III. The author's contribution was the idea and implementation of the *dense component* pattern, also sections such as the inconsistency taxonomy and evaluation.
IV. The author's contribution was the idea and implementation of the pattern of *heart-shaped substructure*, but also the background description of the topic and the overview of the related works.
V. The author's contribution was the idea and implementation of three test patterns applied on four different language wordnets with the description of the results.
VI. The author's main contribution was the idea and implementation of four test patterns with their mathematical models and description of their capability to discover inconsistencies in the semantic hierarchies of wordnet.

LIST OF ABBREVIATIONS AND DEFINITIONS

NLP	natural language processing
WSD	word sense disambiguation
CorWN	Cornetto (Dutch Wordnet)
EstWN	Estonian Wordnet
FinnWN	Finnish Wordnet
PlWN	Polish Wordnet
PrWN	Princeton WordNet

Wordnet semantic hierarchy. In our work, it is mainly a hierarchy where sets of *synonyms* (lexicalized concepts) are in a semantic relation of IS-A or IS-SOME-MANNER.

Synset or **set of synonyms**. A group of cognitively similar *synonyms*. The synonym may be a single word, a compound word, a phrasal verb, a collocation, an idiomatic phrase or a proper noun.

Lexical unit. A member of a *synset* or a *synonym* in a *synset*.

Polysemy. A phenomenon where a word (*lexical unit*) or phrase has two or more meanings, and these meanings are interconnected.

Regular polysemy or **systematic polysemy**. A status where there exist a minimum of two words that have at least two meanings with a similar relation between those meanings.

Multiple inheritance case. A case where one concept in the semantic hierarchy has at least two parents. For instance, the concept {water} may have two parents – {liquid} and {food, nutrient}.

Regularity of multiple inheritance. A status where there exist a minimum of two concepts with at least two identical parents in the semantic hierarchy.

Test pattern. This signifies a class in an object-oriented approach, a description of substructure with a specific nature in the wordnet[1] semantic hierarchy as a graph.

Check. This is used in the context of verification, i.e. we verify the existence of specific structures in wordnet semantic hierarchies.

Validate. In our work *validate* means to inspect whether a substructure with a specific nature in the semantic hierarchy of wordnet fulfils its intended requirements.

[1] hereafter "wordnet" is referred to in lower-case letters as a certain design dictionary or wordnet-type dictionary

INTRODUCTION

"WordNet is extensively used as a major lexical resource in NLP. However, its quality is far from perfect, and this alters the results of applications using it" –
Nervo Verdezoto • Laure Vieu

Computational linguistics utilizes lexical resources in computer-aided semantic analysis (Clark et al., 2013). Often, these resources are text corpora, explanatory dictionaries, but also web-based encyclopedias (Wikipedia) (Gabrilovich and Markovitch, 2009) or common-sense knowledge bases (Cyc (Ramachandran et al., 2005), ConceptNet (Havasi et al., 2009), YAGO (Suchanek et al., 2008), BabelNet (Navigli and Ponzetto, 2012), or DBpedia (Auer et al., 2007)). For over a decade, a trend for creating lexical resources in different languages has been on the rise, in particular of the wordnet-type design which are by their nature hierarchies of lexicalized concepts. These hierarchies are very large and comparable to chip design hierarchies. Wordnet as a synonymy dictionary has quite a different structure from an ordinary or monolingual explanatory dictionary where every entry has as a definition or a description. In wordnet, cognitively similar words are gathered into one set – a set of synonyms or *synsets*. All *synsets* are semantically related, thus composing a forest of hierarchies (Fellbaum, 1998).

Wordnet is typically constructed by expert linguists-lexicographers[2]. However, as wordnet building takes place as a human-machine system, as does chip design, we may expect different types of errors to occur. On the one side, the location of every item in a chip design is in accordance with certain algebra. Wordnet, to the contrary, is not so strictly constrained.

While wordnet hierarchies are very large, it is not very efficient to validate wordnet in an alphabetical word order (Čapek, 2012). Instead, it is reasonable to look at wordnet hierarchies in a general way. For that reason we propose a methodology based on graph theory. Thus, we search specific subgraphs that point to possible errors in this vast semantic hierarchy. We look at these subgraphs as test patterns and use them as descriptions of substructures with a specific nature in order to check their existence in the semantic hierarchies of wordnet. Every instance of a set of test patterns has a different error percentage.

Our approach is not entirely new as different authors have used a graph-based approach to check and validate wordnet as a graph by searching for *cycles* (Šmrz,

[2] Usually, ordinary or non-expert persons can define an ambiguous word with only a few meanings but a linguist-lexicographer with many meanings. For instance, an ordinary Estonian proposed three meanings to the word "tee" – "tee" as tea, "tee" as a road and "tee" as an order to do something. But in Estonian Wordnet, this word has 12 meanings.

2004), (Kubis, 2012), *rings* (Fischer, 1997), (Liu et al., 2004), (Richens, 2008), *dangling uplinks* (Koeva et al., 2004), (Šmrz, 2004), *roots* or *null graphs* (Čapek, 2012) in the semantic hierarchies.

The purpose of this thesis

In the present work we aim to prove that in addition to the abovementioned substructures (*cycles, rings, dangling uplinks, roots and null graphs*), there are other kinds of substructures that are also helpful in the validation of wordnet semantic hierarchies.

More precisely, in this work we study substructures that consist of *multiple inheritance* cases, i.e. the nodes that have many parents and which correspond to the polysemy in the lexical semantics.

In the context of our work, these certain-shaped substructures are called test patterns and substructures found from a particular wordnet with the help of test patterns termed instances.

Motivation

"No two lexicographers have exactly the same knowledge and perspective of a language and that perspective changes even for a single lexicographer over time" – Tomáš Čapek

In the most general sense, we are motivated by the fact that each expanding and developing human-machine system requires a strong feedback control mechanism to evaluate the normal trends of the system as well as the unsystematic steps.

Secondly, in the narrower sense, we are prompted by the fact that the quality of wordnet semantic hierarchies has a strong impact on the quality of natural language processing tasks that use wordnet (Verdezoto and Vieu, 2011).

Thirdly, different lexicographers have different language perceptions which may change over time affecting the construction of the semantic network (Čapek, 2012). Furthermore, our work contributes to linguistics practice in the following ways:

- **Test patterns simplify the work of lexicographers,** thus helping them to check and validate hierarchical structures
- **All given patterns are cross-language**, i.e. these patterns are applicable in all wordnets in the world (there are about 50 different language versions of wordnets)
- Validating all hierarchies at once using test patterns is much quicker than going through hierarchies sequentially (Čapek, 2012)
- **Implemented algorithms** help to find the instances of test patterns.
- An overview of 10 versions from **Estonian Wordnet's iterative evolution** where test patterns were employed may prove the efficiency of test patterns in validation of the semantic hierarchies of this wordnet.

Our work was also driven by reasons related to the feature of *multiple inheritance* cases in test patterns, which refer to possible error(s) in semantic hierarchies:

- 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).
- Sometime IS-A relation is used instead of a different semantic relation (Martin, 2003). *Multiple inheritance* makes it possible to compare relations that connect the various parents of a *synset*.
- Vider (2001) proposes that in Estonian Wordnet every *synset* has in an ideal case only one parent. Test patterns reveal all the different cases.

Further reasons motivating this thesis are given at the beginning of Chapter 3.

Thesis objectives

The objectives of this thesis are to:

- Give background information on how errors occur in wordnet hierarchies (Chapter 1)
- Describe the impact of *polysemy* and *regular polysemy* on the wordnet semantic hierarchy (Chapter 1)
- Give a systematic overview of the validation methods of the wordnet semantic hierarchies and the identification of general errors (Chapter 2)
- Describe and give an overview of the test patterns and the typical errors they may reveal (Chapter 3)
- Provide a numerical overview of the test patterns' instances for 10 versions from Estonian Wordnet's iterative evolution (Chapter 4)
- Create programs for finding instances of test patterns and apply them on some different language wordnets (Chapter 5)

Research objects

The central objects of this research are the substructures of semantic hierarchies of wordnet that consist of possible flaws in the noun and verb hierarchies. Henceforth, the nodes of these hierarchies are described as sets of synonyms or *synsets* (or lexicalized concepts) that group words with similar meanings. The edges represent *hypernymy* relations, which in the case of noun hierarchy correspond to IS-A or IS-KIND-OF relations and in the case of verb hierarchy to IS-SOME-MANNER/WAY relations.

Methodology

The main research method in this dissertation is pattern-based validation. We use a methodology which is divided into two phases of action:

- On the basis of test patterns our programs find their instances for any wordnet version
- A lexicographer validates the instances and corrects them in the management system of wordnet, if necessary.

Source information

The central source in this research is a wordnet, a lexical-semantic database. In this context, wordnet is used as a semantic hierarchy.

Mostly, we use Estonian Wordnet versions in the range of 60 to 70, but Princeton WordNet (versions 3.0 and 3.1), Finnish Wordnet (FinnWordNet version 2.0), Dutch Wordnet (Cornetto version 2.0) and Polish Wordnet (plWordNet versions 1.8 and 2.0) are also used.

Novelty

The theoretical novelty lies in the new test patterns being presented as graphs and highlighting *multiple inheritance* cases in the semantic hierarchy.

The practical novelty lies in the algorithms implemented for finding instances of test patterns. Secondly, instances of test patterns are used as a lexicographer's tool for validating the semantic hierarchy.

RESEARCH QUESTIONS

How to check and validate the wordnet semantic hierarchy?

(Chapter 1) How different construction approaches may affect the wordnet semantic hierarchies?

- What is the impact of the particular feature of *hypernymy* on the semantic hierarchy?
- What is the impact of polysemy on *multiple inheritance* in the wordnet semantic hierarchy?
- What is the impact of *regular polysemy* on the *regularity of multiple inheritance*?
- What are the three aspects every wordnet creator has to consider and how do they affect the quality of the wordnet semantic hierarchies?

(Chapter 2) What methods are used in the validation of the semantic hierarchies of a wordnet?

- How to systematize the methods of validating employed in semantic hierarchies?
  - What kind of features are used to classify them?
  - Into which group of methods does our approach belong?
- What types of errors occur in wordnet?
  - Into which group of methods does our approach belong?

(Chapter 3) What test patterns to use in order to check and validate the semantic hierarchies of a wordnet?

- How to describe test patterns?
- What is the most similar work to ours?
- What direction to follow in validating on the basis of different test patterns?

- What kinds of errors are typical to every test pattern?

(Chapter 4) How to validate the instances of test patterns in the wordnet semantic hierarchies in practice?

- What are the examples for validating the instances of test patterns?
- Who validates the semantic hierarchies of a wordnet?
- When to validate the semantic hierarchies of a wordnet?

(Chapter 5) How to check the instances of test patterns in the wordnet semantic hierarchies?

- What main actions of a program are to be implemented to find instances of test patterns?
- What is the effect of the test patterns used for validation on the EstWN semantic hierarchies?

DISSERTATION OUTLINE

Chapter 1 provides background information about wordnet including its design, applications, and the basic principles of wordnet hierarchical structure and three aspects of wordnet construction. The conclusions section points out how these aspects of construction – lexical resources, building models and automation levels – give many opportunities to import errors into the semantic network of wordnet.

Chapter 2 focuses on the validation methods used in the semantic hierarchies of wordnet. This chapter divides validation methods into three groups and introduces them in decreasing order of their popularity. In addition, it gives an overview of the three type of errors with examples.

Chapter 3 introduces a validation method based on a system of test patterns proposed by the author of this thesis. All test patterns in this system are described as graphs and associated with the typical semantic errors they may help to discover.

Chapter 4 puts test patterns into practice, demonstrating the usage of test patterns' instances in the validation of Estonian Wordnet. The author of this thesis[3] describes examples of each instance and points in certain cases to the differences with the latest wordnet version. Also, a case study, validated by a lexicographer, of the *dense component*'s test pattern is presented.

Chapter 5 introduces the main actions of the programs implemented by the author to find test patterns' instances. Additionally, it includes an overview of the Estonian Wordnet's iterative evolution that is based on versions 60 to 70 and demonstrates

[3] Ordinarily, this would be performed by a lexicographer

how the use of test patterns affects the wordnet structure. The condition of semantic hierarchies based on the number of test patterns' instances is also presented for four other wordnets – Princeton WordNet, Finnish Wordnet, Dutch Wordnet, and Polish Wordnet.

The **final chapter** summarizes the results of this dissertation and presents plans for future work.

1. THEORETICAL BACKGROUND

"A wordnet is a computerized dictionary of synonyms, thesaurus, lexical database, taxonomy of concepts – the list can go on." –
Maciej Piasecki • Stanisław Szpakowicz • Bartosz Broda

This chapter provides some background information for understanding the nature of a wordnet and the topics related to the construction of the semantic hierarchies of wordnet.

Firstly, a wordnet is defined, its design described and its popularity as lexical knowledge source in natural language processing (NLP) is demonstrated.

Secondly, the basic principles of wordnet hierarchical structure are covered, including topics like *polysemy*, *regular polysemy*, *multiple inheritance* and the *regularity of multiple inheritance*, which all have an essential impact on the wordnet hierarchy.

Finally, three factors are described that must be taken into account in the construction of a wordnet. These are the lexical resource(s), building model and automation level. Depending on which lexical resource, model or automation level is used in the building or expanding of a wordnet, they may have a big impact on importing errors into this semantic network.

1.1 ABOUT WORDNET

Wordnet is a lexical-semantic database often used as background knowledge source in natural language processing applications (Fellbaum, 1998a), (Reynaud and Safar, 2007). In addition to the given definition, wordnet is also described as a computerised dictionary of synonyms, a taxonomy of concepts, a thesaurus, or a lexical ontology (Piasecki et al., 2009), (Gómez-Pérez and Benjamins, 1999).

This section now turns to the history of wordnet, its applications and design.

1.1.1 A short history

According to (Fellbaum, 2010) the WordNet project started in 1986 at Princeton University and was headed by George A. Miller. Similarly to the work of (Collins and Quillian, 1969), Miller as a psycholinguist was interested in how the human semantic memory is organized. The model proposed by Collins and Quillian prefigures hierarchically structured concepts as seen in Figure 1.1.

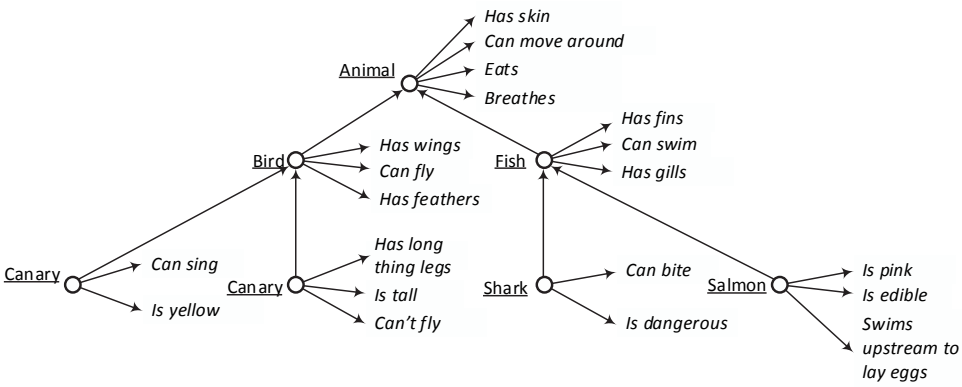


Figure 1.1 *Illustration of a hypothetical memory structure for a 3-level hierarchy. Figure originates from (Collins and Quillian, 1969)*

In their hierarchy, concepts that are more specific inherit information[4] from more general concepts. Moving in a hierarchy from top to down, every concept stores information that is specific only to itself. According to Figure 1.1, for "Fish" is the specific features are "Can Swim", "Has fins" and "Has gills", whereas for "Shark" these are "Can bite" and "Is dangerous". Information, which is not specific, "Shark" inherits from the concept of "Fish".

Miller and his team intended to represent the lexicalized concepts of a language "*with a hierarchical structure in a network-like structure*". The result of their work was a vast, manually constructed semantic net – WordNet[5]. Its aim is no longer to model the human semantic memory. Instead, it has become a most used/useful tool for NLP and in lexical semantics and ontology research (Fellbaum, 2010).

1.1.2 The mother of all wordnets

WordNet or Princeton WordNet has "*become a synonym of a particular kind of lexicon design*". Since other wordnets follow a design similar to the Princeton WordNet, it is also referred to as *the mother of all wordnets* (Fellbaum, 1998b). Today, there are more than 70 wordnets in the world in about 50 languages. According to the web page of Global Wordnet[6], all of these 70 wordnets include links to WordNet or to others that have links to Princeton WordNet.

Many wordnets have been developed under a multilingual wordnet project. For instance, after Princeton WordNet, *EuroWordNet* project commenced in March 1996. It contains wordnets for European languages such as Dutch, Italian, Spanish, German, French, Czech and Estonian (Vossen, 1998a). *The BalkaNet* project

[4] It includes properties (e.g. *Has skin*) as well acts (e.g. *Can move around*)
[5] WordNet is a registered trademark, owned by Princeton University
[6] http://globalwordnet.org/wordnets-in-the-world/

developed wordnets for Czech and five Balkan languages – Bulgarian, Greek, Romanian, Serbian, and Turkish (Tufis et al., 2004a). *The IndoWordNet* project developed 18 wordnets for Indian languages (Bhattacharyya, 2010). In addition, there are other multilingual wordnet projects – *Open Multilingual Wordnet* (it includes Arabic, English, Malaysian, Indonesian, Finnish, Hebrew, Japanese, Persian, Thai, and French), *Asian WordNet* (it includes Hindi, Indonesian, Japanese, Lao, Mongolian, Burmese, Nepali, Sinhala, Thai and Vietnamese) (Charoenporn et al., 2008) and others. Yet, there are many wordnets not developed under any multilingual wordnet project. For example PersiaNet (Montazery and Faili, 2010), FinnWordNet (Lindén and Niemi, 2014), RussNet (Azarova et al., 2002), PolNET (Vetulani et al., 2010) and plWordNet (Maziarz et al., 2012). In addition, there exist wordnets, which started under a multilingual wordnet project but later developed individually, e.g. Estonian Wordnet (Kerner et al., 2010).

1.1.3 Wordnet applications

*WordNet ... has found **myriad applications** in the field of natural language processing* – *Tony Veale*[7]

An "ordinary man" may use wordnet as a synonyms dictionary, by entering into the wordnet search field a word to which he/she is looking for synonyms or by checking the various meanings of ambiguous words. For such use, there are about 30 wordnets with an online browsing facility, including Estonian Wordnet[8], FinnWordNet[9], Hindi WordNet[10], plWordNet[11], Princeton WordNet[12] and sloWNet[13].

Many papers refer to wordnet as a lexical background resource or a background knowledge base for NLP tasks, but the most highlighted task is ***word sense disambiguation*** (WSD) – "*conventionally regarded as the task of identifying which of a word's meanings (senses) is intended, given an observed use of the word and an enumerated list of its possible senses*" (Resnik and Lin, 2010). The role of wordnet in that task is to find out the right sense of an ambiguous word. WSD in turn may be a subtask for *machine translation*, *query expansion*, *information retrieval*, *conceptual identification*, *semantic distance* et al.

[7] http://www.odcsss.ie/node/39
[8] http://www.cl.ut.ee/ressursid/teksaurus/teksaurus.cgi.et
[9] http://www.ling.helsinki.fi/cgi-bin/fiwn/search
[10] http://www.cfilt.iitb.ac.in/wordnet/webhwn/wn.php
[11] http://plwordnet.pwr.wroc.pl/wordnet/
[12] http://wordnetweb.princeton.edu/perl/webwn
[13] http://nl.ijs.si/slowtool/

Wordnet in the composition of other knowledge resources

In order to expand the application field of wordnet and its capability, it is mapped to ontologies, e.g. SUMO (Niles and Pease, 2003) and DOLCE (Gangemi et al., 2002a). In addition, it has been a source for building huge new knowledge bases, e.g. YAGO (Suchanek et al., 2008) and BabelNet (Navigli and Ponzetto, 2010).

YAGO is "a light-weight and extensible ontology with high coverage and quality"(Suchanek et al., 2007). It has 1.7 million entities (with 15 million facts about entities) and relations automatically acquired from Wikipedia and WordNet. After the leveraging of YAGO with Multilingual WikiPedias, the authors of (Mahdisoltani et al., 2015) developed YAGO3 – a multilingual knowledge base with 10 million entities and 120 million facts about entities and a database of GeoNames[14].

BabelNet is "*a very large, wide-coverage multilingual semantic network*" with high quality (Navigli and Ponzetto, 2010). Similarly to YAGO, BabelNet has an automatically made construction that uses lexicographic knowledge from WordNet and encyclopaedic knowledge from Wikipedia.

While WordNet does not consist of many names, YAGO and BabelNet compensate considerably for this drawback and allow to enquire from the knowledge base after "*name entities like people, organizations, geographic locations, books, songs, products, etc., and also relations among these such as whatis-located-where, who-was-born-when, who-has-won-whichprize*" and others (Suchanek et al., 2007).

In addition to YAGO and BabelNet, there are other experiments that have aligned WordNet to other knowledge bases. For example, aligning Wordnet-Wikipedia-Wiktionary (Miller and Gurevych, 2014), WordNet-FrameNet (Baker and Fellbaum, 2009) and WordNet-VerbNet-FrameNet-PropBank (de Lacalle et al., 2014).

Domain/topic wordnets

SentiWordNet is a lexical resource for *opinion mining* or *sentiment analysis*. SentiWordNet is based on WordNet, where each *synset* is supplied to three numerical scores, describing how objective, positive, and negative the *lexical units* contained in the *synset* are (Esuli and Sebastiani, 2006).

Q-WordNet is a lexical resource for *opinion mining* or *sentiment analysis*. It consists of a subset of WordNet senses classified as positive or negative (Agerri and García-Serrano, 2010).

Jur-WordNet is „*an extension for legal domain of the Italian ItalWordNet database, aimed at providing a knowledge base for the multilingual access to sources of legal information*" (Sagri et al., 2004).

[14] GeoNames is a geographical database; http://www.geonames.org/

Medical WordNet is "*a free-standing lexical database designed specifically for the needs of natural-language processing in the medical domain and*" it uses medical terms from WordNet (Smith and Fellbaum, 2004).

Geo-WordNet is a Princeton WordNet in which geographical entities have their coordinates. It is useful for the related tasks of Geographical Information Retrieval (Buscaldi and Rosso, 2008).

WordNet in web applications

Visual Thesaurus[15], Visuwords[16], WordVis[17], JavaScript Visual Wordnet[18] are all web-based visual dictionaries that visualize the semantic net for entered words (Vercruysse and Kuiper, 2013). They primarily use IS-A relation. These tools only employ the lexical resources of Princeton WordNet.

The Free Dictionary[19] is "*an American online dictionary and encyclopaedia that gathers information from a variety of sources*" including Princeton WordNet (Farlex, 2009).

ImageNet[20] "*is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images*" (Deng et al., 2009).

Wordnik[21] is possibly the biggest (social) online dictionary of English. It uses different dictionaries and encyclopaedias for word definitions and example sentences from news sites and blogs. Wordnik is the place for looking up any and every word and it has the support of the social community (Davidson, 2013).

Synonym[22] is an online dictionary that proposes to the user synonyms, antonyms and definitions for the entered word.

Software packages

WordNet::Similarity is a Perl package "*to measure the semantic similarity and relatedness between of concepts (or synsets)*". It supports the measures of Resnik, Lin, Jiang-Conrath, Leacock-Chodorow, Hirst-St.Onge, Wu-Palmer, Banerjee-Pedersen, and Patwardhan-Pedersen (Pedersen et al., 2004).

RiTa.WordNet is a Java library that provides among other things the distance metrics between ontology terms (Howe, 2009).

[15] https://www.visualthesaurus.com/
[16] http://www.visuwords.com/
[17] http://wordvis.com/
[18] http://kylescholz.com/projects/wordnet/
[19] http://www.thefreedictionary.com/
[20] http://www.image-net.org/
[21] https://www.wordnik.com/
[22] http://www.synonym.com/

1.1.4 Wordnet design

The central building block of wordnet is the set of *synonyms* or **synset** (also called lexical concept) (Figure 1.2 and Figure 1.3) which groups together cognitively similar *synonyms*. The *synonym* may be *a single word, compound word, phrasal verb, collocation, idiomatic phrase* or *proper noun*. More formally, a *synonym* in a *synset* is referred to as a *lexical unit*. Between *lexical units* there are **lexical relations** – typically a *synonymy* or *antonymy* (Figure 1.3). Between *synsets* there are **conceptual-semantic** relations[23]. Some conceptual relations form hierarchical structures (*hypernymy*, *holonymy*) and others non-hierarchical structures (*near-synonymy*[24], *role*) (Figure 1.4). The most important semantic relation in wordnet is the *hypernymy* relation, also known as the IS-A relation, which forms a conceptual taxonomy. Every *synset* has information about its **identifier**, part-of-speech **(POS)**, **gloss** and sometimes **usage examples** (Figure 1.2). There may be additional information about the **domain category**. If a *synset* contains a polysemous word, then its parents (in a *hypernymy* or *holonymy* relation), gloss, usage examples and domain category help to identify its meaning.

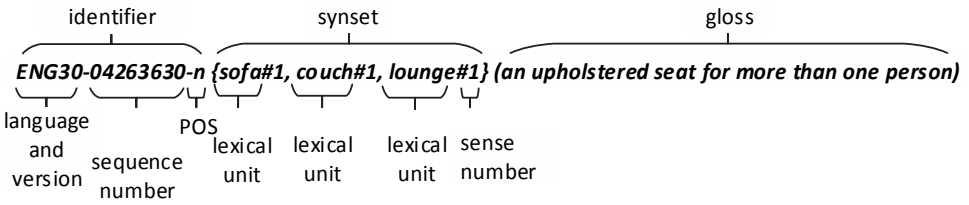


Figure 1.2 *An example of a synset*

The conceptual model in Figure 1.3 represents concepts and their interrelationships in wordnet as a system. This model generalizes all the wordnets in the world. Based on personal experience, not every wordnet uses the concepts of *Domain category, Gloss, Usage example, Interlingual Index* as is apparent in Figure 1.3. Interlingual Index or ILI is the equivalence relation that refers to a *synset* in Princeton WordNet. In IndoWordNet, ILI is a reference to a *synset* in Hindi WordNet.

[23] Although, in a broader meaning *lexical relations* as well as *conceptual-semantic relations* are both *semantic relations*. Hereafter, we always refer to word specific relations as *lexical relations* and to *concept specific* relations as *semantic relations*.

[24] *near-synonymy* = almost with the same meaning or almost synonym

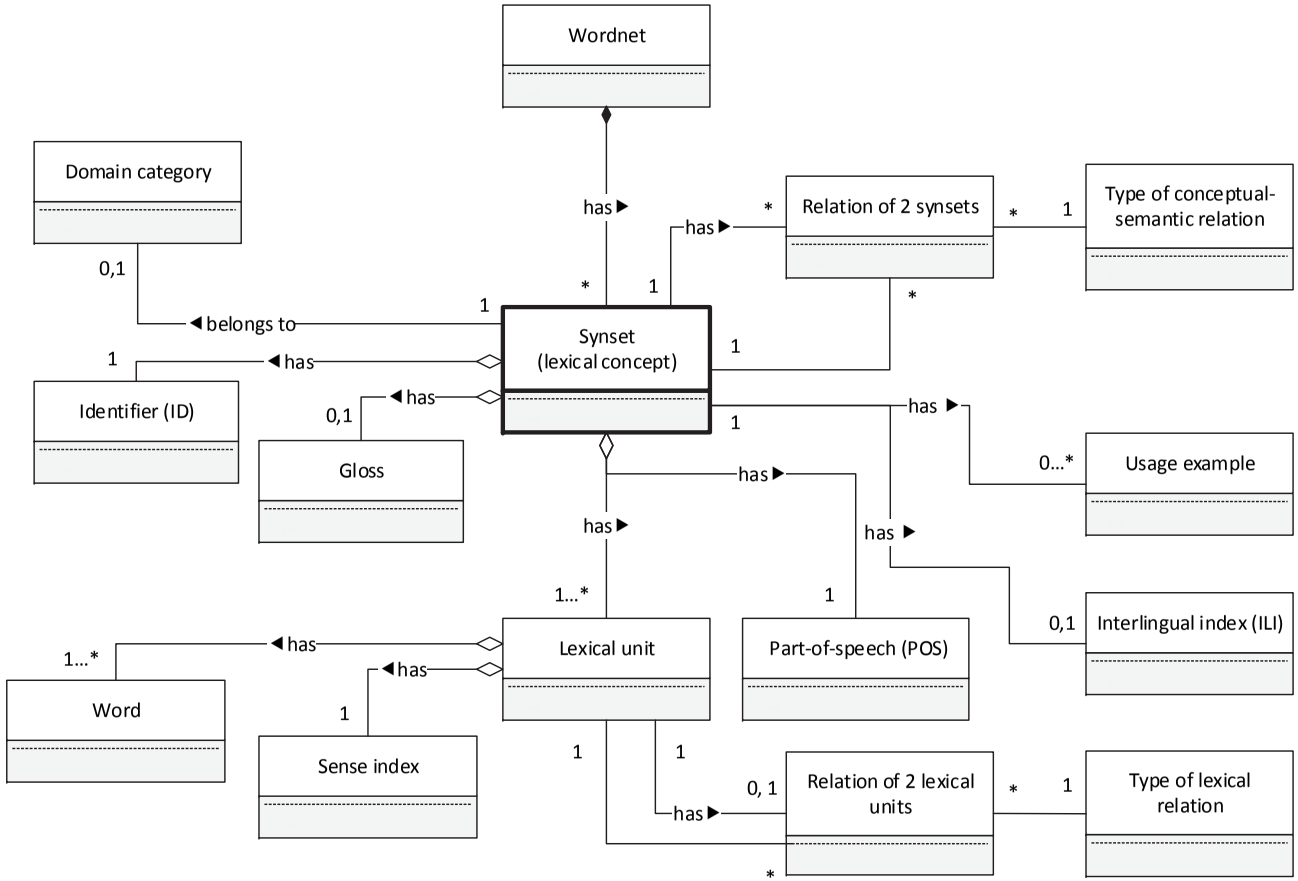


Figure 1.3 *Conceptual model of wordnet*

Table 1.1 *Explanation to the conceptual model of wordnet*

Name	Semantics
Synset (= set of synonyms)	Lexicalized concept; consists of words with cognitively the same meaning; a central building block of wordnet
Lexical unit	Member of a *synset*; *lexical unit* may be a *single word*, *compound word*, *phrasal verb*, *collocation*, *idiomatic phrase or proper noun*
Word	Member of the *lexical unit*
Sense index	Sequence number that helps to distinguish *lexical units* with multiple meanings, the type of value is an integer; initial value is 1
Relation of 2 lexical units	
Type of lexical relation	Typically a *synonym* or an *antonym* relation (see Figure 1.4)
Part-of-speech	Syntactic category, usually a *noun*, *verb*, *adjective or adverb*
Gloss	Definition of a *synset*; typically, it is given in monolingual or explanatory dictionaries for an entity
Identifier	Unique identifier of the *synset*; it may consist of a sequence number and sometimes an abbreviation of the language and/or wordnet version and/or part-of-speech; For example: *ENG30-04047401-n* (Princeton WordNet, version 3.0), *d_n-12651* (Cornetto, version 2.0)
Domain	Every *synset* belongs to one domain; domain examples: *communication*, *time*, *body*, *act*, *artefact* (from Princeton WordNet version 3.1)
Relation of 2 synsets	
Type of semantic relation	Typically *hyponymy*, *hypernymy*, *part meronymy*, *part holonymy*. There are about 30 relations in Princeton WordNet and 40 in Estonian Wordnet
Usage example	Sentence where the *lexical unit* of a *synset* is used
Interlingual Index	The equivalence relation that refers to the Princeton WordNet *synset*

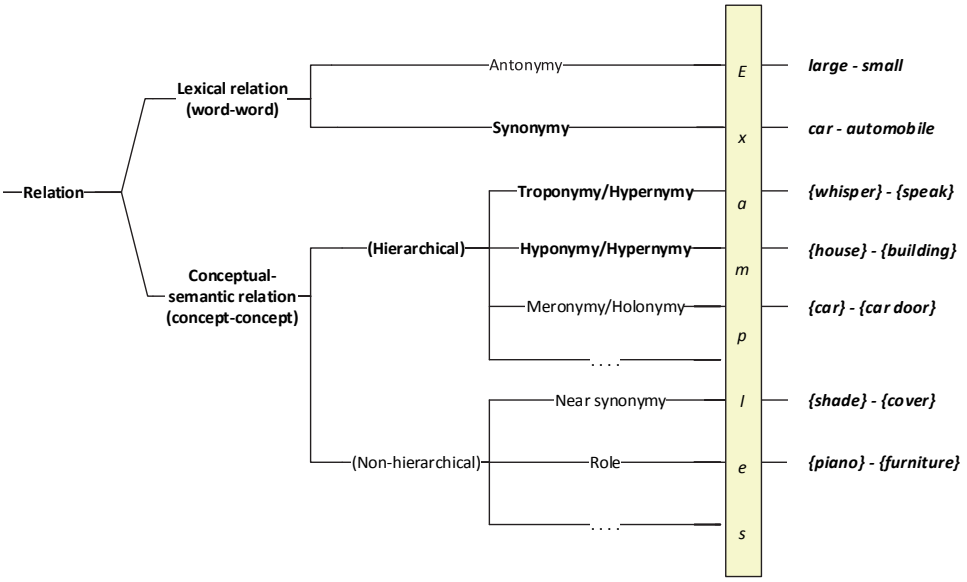


Figure 1.4. *Relations in the structure of wordnet*

1.2 WORDNET HIERARCHY

As the aim of this thesis is to deal with noun and verb *synsets* as well as *hypernymy/hyponymy* and *hypernymy/troponymy* relations, wordnet hierarchy is only described from that perspective. Here, *hypernymy/hyponymy* belong together with noun *synsets* and *hypernymy/troponymy* goes together with verb *synsets*. They both form freestanding hierarchies. This kind of noun hierarchy may run 15 or more levels deep. On the contrary, verb hierarchies are rather flat and "bushy" (Figure 1.5, Figure 1.6) and usually have an up to 5-level hierarchy (Fellbaum, 2002a).

While every hierarchy presupposes the existence of a *root synset* or a unique beginner, this section begins with the top concepts. In addition, the principles of constructing a *synset* are examined, followed by the principles of constructing semantic relation. It should be noted that even though *hypernymy/hyponymy* and *hypernymy/troponymy* relations connect more general *synsets* to more specific ones, they have entirely different semantic foundations.

1.2.1 Top concepts

An essential indicator that determines the wordnet hierarchy is the unique beginners or top concepts, i.e. *synsets* (or according to the previews section, a lexicalized concept or just a concept) with at least one subordinate and no superordinate. These unique beginners divide a particular part-of-speech into several hierarchies, each with a different *unique beginner* or *root synset* or also a *primitive semantic component*. These hierarchies may thread to each other and vary widely in size.

Obviously many wordnet researchers agree that the number of unique beginners cannot be left to chance. However, there is a problem: how to discover these unique beginners? Even if the different researchers have diverse opinions, it is critical to follow a particular criterion, namely, every word of a particular part-of-speech in the target language must have a place in these hierarchies (Miller, 1998). George Miller, the main creator of the first wordnet, claims that in the case of **noun hierarchy's** disjunction Philip N. Johnson-Laird's analysis was contemplated and **25** unique beginners "*were selected after considering the possible adjective-noun combinations that could be expected to occur*" (Miller, 1998). All 25 hierarchies combined cover distinct conceptual and lexical domains. Twelve of them are as follows: *{act, activity}*, *{animal, fauna}*, *{artefact}*, *{attribute}*, *{body}*, *{cognition, knowledge}*, *{communication}*, *{event, happening}*, *{feeling, emotion}*, *{food}*, *{group, grouping}*, *{location}*. Later, according to (Miller and Fellbaum, 2007) there were repeated requests to merge all these 25 initial hierarchies. Due to this, Princeton WordNet now provides **a single** noun *root synset {entity}*.

There is only one large hierarchy of nouns in Princeton WordNet, the top hierarchy of which is represented in Figure 1.5. Every node denotes a *synset*. The number in brackets shows number of *synset* subordinates.

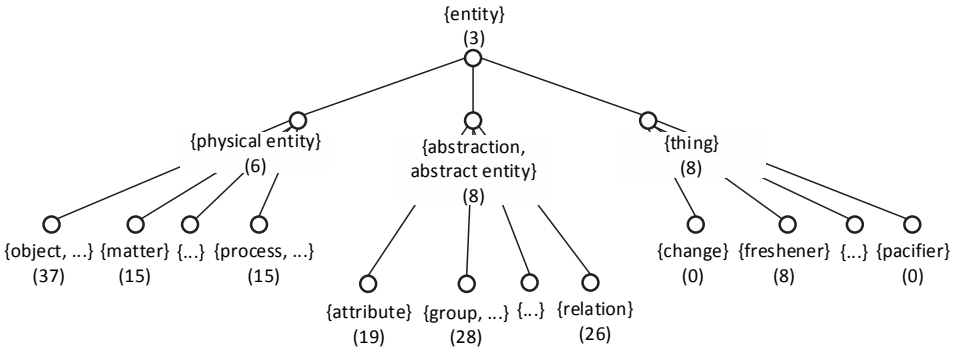


Figure 1.5. *Top hierarchy of noun, Princeton WordNet (version 3.1)*

In spite of this Fellbaum (1998c) does not reveal the method they used to choose unique beginners for **verb hierarchies** after a discussion on which unique beginners could be suitable; she confirms that they settled on unique beginners for the **14** semantic domains. Later, Fellbaum proposed (personal communication, 17.01.2013) a verb hierarchy with only three unique beginners: *to be*, *to do*, *happen*.

In Figure 1.6, a verb hierarchy with the unique beginner "be" is represented. As we speculate based on this figure, the maximum depth of this hierarchy is four. In the case of verb hierarchies, it is quite common.

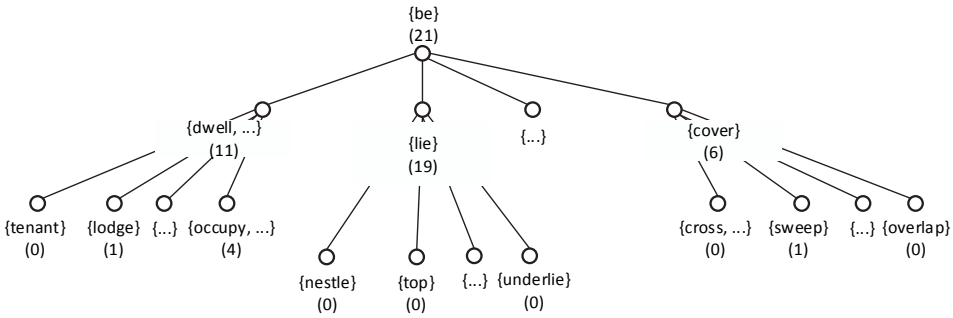


Figure 1.6. *Top hierarchy of verb, Princeton WordNet (version 3.1)*

1.2.2 Principles of constructing a synset

A *synset* as a building block of wordnet is "*a group of cognitively synonymous words*" that defines a particular sense uniquely (Ramanand and Bhattacharyya 2008) (Fellbaum, 2002b). Constructing a wordnet *synset*, three principles have to be adhered to: a) minimality, b) coverage and c) interchangeability (or replaceability) (Miller, 1998)

The minimality principle requires that all and only *synonyms* should be *synset* members (e.g., {world, Earth, earth, globe} – the third planet from the sun; the planet we live on).

The coverage principle means that a *synset* must consist of all the words that represent a particular meaning in this language (e.g., {world, human race, humanity, humankind, human beings, humans, mankind, man} – all of the living human inhabitants of the earth).

According to Miller, the **interchangeability principle** in a *synset* is presupposed rather than asserted. That is to say, *synonyms* in the wordnet *synset* are interchangeable only in certain circumstances (Miller, 1998).

1.2.3 Principles of constructing semantic relations

Semantic relations in wordnet are relations between *synsets* (concepts). The most significant relationship from the perspective of wordnet applicability in different NLP applications is the semantic relation of *hypernymy*. *Hypernymy* relation is a transitive, asymmetric and generalization relation. The inverse relation to *hypernymy* is *hyponymy* in the case of noun *synsets* and *troponymy* in the case of verb *synsets*. This inverse relation is also a transitive, asymmetric but a specialization relation. The compiling for noun and verb *synset* by a *hypernymy* relation takes place on different bases. This difference also occurs in their organisation, thereby the organisation of verbs is more complicated than that of nouns. In addition, not all verbs can be gathered under a single top node (Fellbaum, 1998c), (Fellbaum, 2002a) and (Lo et al., 2008).

1.2.4 Basis of hyponymy/hypernymy relation

Hypernymy relation regarding **nouns** can be read '**is-a**' or '**is-a-kind-of**'. Formally, if two *synsets* {A} and {B} are given and have a *hypernymy* relation between them, then:

{B} is *hypernym* of {A} if {A} is-a (kind-of) {B}

{B} is *hyponym* of {A} if {B} is-a (kind-of) {A}

For *hyponymy* definition, Miller proposes an approach that is more detailed. He posits it as follows: "*when the features characterizing synset {A} are all included among the features characterizing synset {B}, but not vice versa, then {B} is a hyponym of {A}*". This approach defines the *hyponym* through its features. Thus, to find out if one *synset* is the *hyponym* of another, one must check whether one list of characteristics is included in another one.

Later, Miller refers to the work of Wierzbicka (who distinguishes five kinds of *hypernymy* relations) (Wierzbicka, 1984) and confirms that noun *hypernymy* relation "*represents actually more than one semantic relations*". Miller believes that "*two of them seem are particularly salient*". One is the abovementioned 'is-a-kind-of' relation and another is the 'is-used-as-a-kind-of' relation. Wierzbicka associates them with "taxonomic" and "functional" categories/concepts. She uses examples, for instance a *bird* is a **taxonomic category**/concept for *swallow* or *parrot*, and a toy is a **functional category**/concept for a tricycle. Instead of taxonomic and functional categories, Pustejovsky uses notions of "**formal**" and "**telic**" roles (Pustejovsky, 1991).

Hypernymy relation	**Wierzbicka**	**Pustejovsky**	**Examples**
'is-a-kind-of'	Taxonomic category	Formal role	{chicken} → {bird}
'is-used-as-a-kind-of'	Functional category	Telic role	{chicken} → {food}

In the opinion of (Miller, 1998), a *hypernym* may sometimes be purely formal, or purely telic but there are also complicate cases where a *hypernym* is both formal and telic. These three cases provide three different approaches to dealing with these situations. In addition, one can see that these distinctions determine the character of the hierarchical structure. Following examples originate from Princeton WordNet (version 3.1).

1) The *hypernymy* relation represents both a formal and a telic relation, one *hypernym*:
   {poker} → {fire iron}
2) *Hypernymy* relations represent both a formal and a telic relation, two or more *hypernyms*:
   {water} → {liquid} (formal)
   ↘ {food, nutrient} (telic)
3) *Hypernymy* relations represent both a formal and a telic relation, two *hypernyms*:
   {chicken} → {bird}
   {chicken} → {food}

1.2.5 Basis of troponymy/hypernymy relation

Hypernymy relation regarding **verbs** can be read as '**manner-of**'. According to (Fellbaum, 1998c), the sentence frame *An X is a Y* for testing the *hyponymy* relation between nouns is not suitable for verbs. If two verbs V_1 and V_2 are given, then the *troponymy* (also manner) relation between these two verbs might be expressed as follows:

To V_1 is to V_2 in some manner/way (Fellbaum, 2002b).

V_1 is a more specific verb than V_2, as was true about the *hyponymy* relation of the noun. However, the comparison approach for whether the features of one word are included in another one is not appropriate here. ***Troponymy* is a particular kind of entailment**. More precisely, there is a *troponymy* relation between two words if:

1) that pair is always temporally co-extensive and
2) is related by entailment

Thus, ***V_1 entails V_2*** represents the ***troponymy*** relation if and only if ***V_1 and V_2 are simultaneously coextensive*** (Fellbaum, 1998c).

In her explanation, Fellbaum uses the example of "whisper-speak". *To whisper is to speak in some manner. Whispering* entails *speaking* and they both are coextensive.

1.2.6 Lexical ambiguity

Lexical ambiguity is a fundamental problem in natural language processing tasks (Krovetz, 1997) but also has a high impact on the wordnet hierarchical structure. For example, ambiguous words with their related meanings form different clusters in the wordnet hierarchy (Section 1.2.9). Moreover, there is no written rule on how to organize ambiguous words with a similar meaning in the wordnet hierarchy (Verdezoto and Vieu, 2011). Is it one *synset* with many parents or many *synsets* with one parent?

The following example is a simplification where a word represents a *synset*. According to Figure 1.7, there are two different types of lexical ambiguity – *homonymy* and *polysemy*.

Homonymy is a phenomenon where "*one of two or more words [are] spelled and pronounced alike but [are] different in meaning (as the noun quail and the verb quail)*"[25].

Polysemy is a phenomenon where a word (*lexical unit*) or phrase has two or more meanings, and these meanings are interconnected (Langemets, 2010). According to (Apresjan, 1974), the polysemy definition does not require that there is a common part of all the meanings of a polysemic. It is enough that each meaning has at least one link to the other one to which it has a related meaning.

[25] http://www.merriam-webster.com/dictionary/homonym

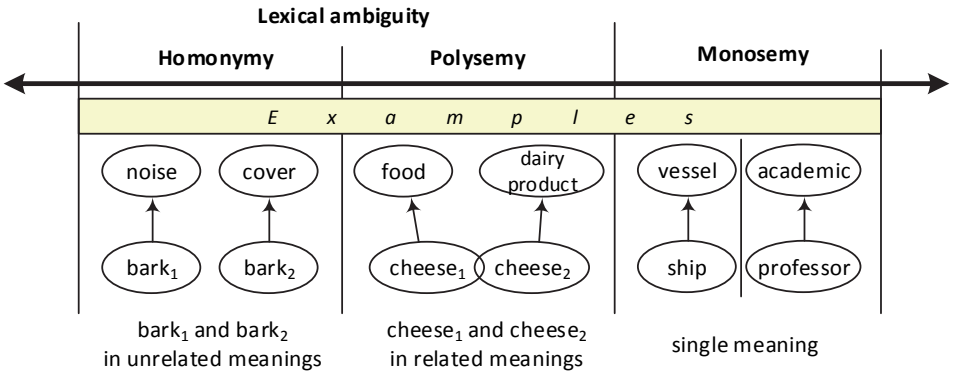


Figure 1.7. *The homonymy-polysemy-monosemy axis*[26]

The difference between a homonym and a polysemic word is in whether the words have a *related meaning*[27]. Thus, in both cases, one word has several different meanings but in the case of homonymy, the meanings are unrelated. In the event of polysemy, they have related meanings (Figure 1.7, Figure 1.8). In addition, if a word is polysemic, then typically it has a central sense and subsenses (Langemets, 2010).

In our work, the main focus is on the concept of polysemy because homonyms do not form any specific structures in the wordnet semantic hierarchy because they are separated from each other. Figure 1.8 represents a more precise classification of polysemy. This diagram is based on the description of (Freihat et al., 2013), where the authors used a different view of polysemy. They classified it as *complementary* and *contrastive polysemy*. *Complementary polysemy* corresponds to *polysemy* in our work, and *contrastive polysemy* corresponds to *homonymy*.

[26] The idea of the homonymy-polysemy-monosemy axis comes from (Pethö, 2001)

[27] More precisely, *etymologically related meaning*

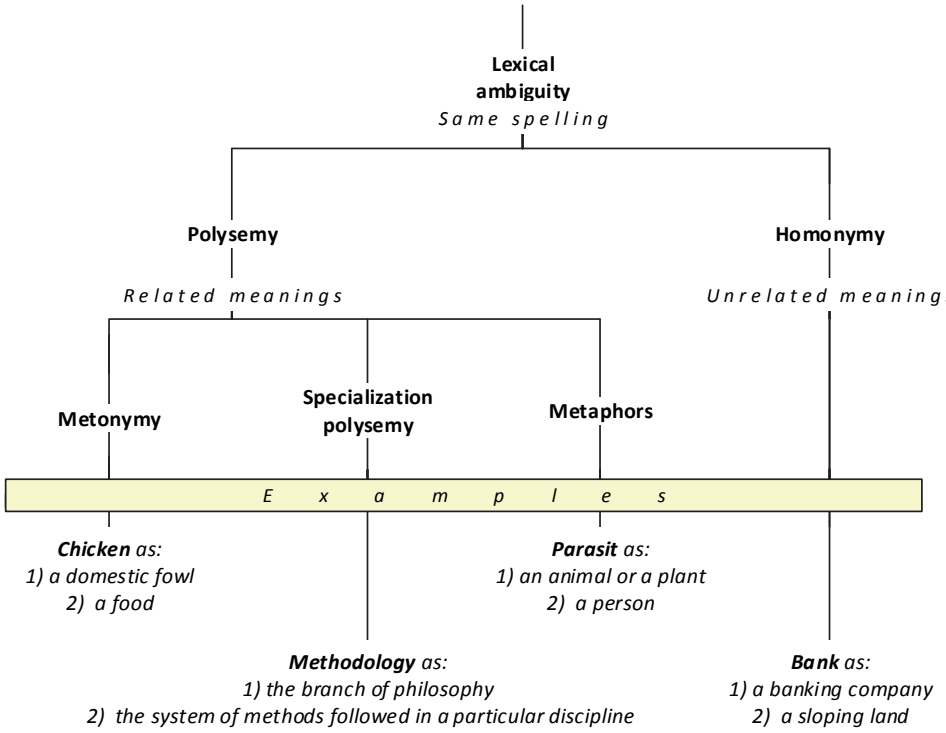


Figure 1.8. *Classification of lexical ambiguity*

According to (Freihat et al., 2013) (Figure 1.8), it is possible to classify polysemy into three sub-concepts – *metonymy*, *specialization polysemy*, and *metaphors*. The authors of (Freihat et al., 2013) created relations between polysemous words in the wordnet noun hierarchy to indicate the specific type of polysemy. They found that metonyms and metaphors were located on the top level of the wordnet hierarchy. Specialization polysemy was situated on the middle as well as the lower level of the wordnet hierarchy.

Metonymy is the use of a word or a phrase in a figurative sense on the basis of a chronological, spatial, causal or other relation[28].

Specialization polysemy is a word or phrase used "*to refer [to] a more general meaning and a more specific meaning*" (Freihat et al., 2013).

Metaphor is "*a word or phrase for one thing to refer to another thing in order to show or suggest that they are similar*"[29].

[28] Translation from Estonian Explanatory Dictionary: http://www.eki.ee/dict/ekss/

[29] http://www.merriam-webster.com/dictionary/metaphor

1.2.7 Polysemy vs. multiple inheritance

As discussed at the beginning of this section, there are two ways to add polysemous words into the wordnet hierarchical structure (Verdezoto and Vieu, 2011). The first method generates the *multiple inheritance* but the second one does not. *Multiple inheritance* in wordnet hierarchies refers to a case where one *synset* has at least two parents, and thus that the *synset* inherits properties from many concepts.

Figure 1.9 depicts a polysemous "cheese" with the meanings "food" and "dairy product". Case 1 presents the first way to add polysemous "cheese" into the wordnet hierarchy. In that event, "cheese" is in the wordnet hierarchy in two separate *synsets* with senses 1, 2. Additionally, both senses only have one parent (*hypernym*).

In Case 2, polysemous "cheese" appears in one *synset*, and it simultaneously has two parents – {food} and {dairy product}. In other words, despite the fact that "cheese" occurs only in one *synset* of wordnet, it has, in fact, two senses.

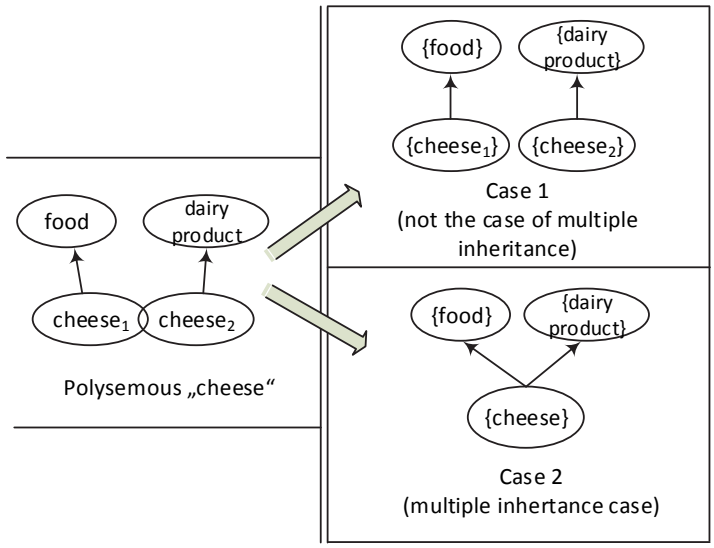


Figure 1.9. *Polysemy vs. multiple inheritance*

1.2.8 Regular polysemy vs. the regularity of multiple inheritance

(Langemets, 2010) refers to the work of (Apresjan, 1974) and notes that based on Russian examples Apresjan proved that in large part, polysemy is not occasional, but it quite regularly follows patterns, which indicate *regular polysemy*.

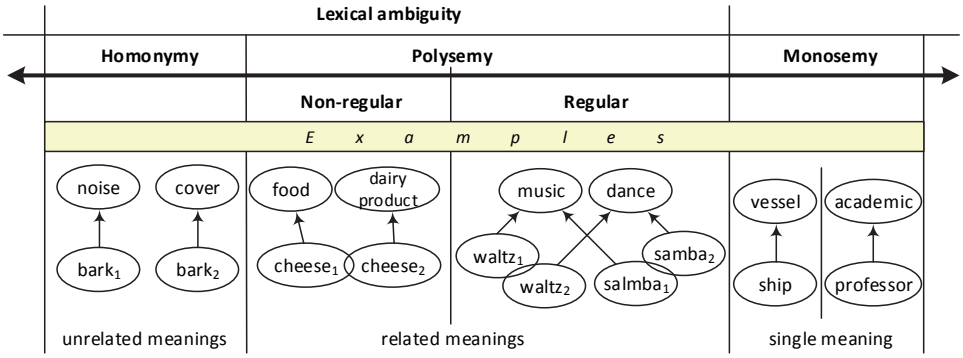


Figure 1.10. *Regular and non-regular polysemy*[30]

(Langemets, 2010) defines *regular polysemy* as a status where a minimum of two words have at least two meanings with a similar relation between those meanings. For example, if the word *waltz* means both *music* and *dance*, then the same is true about *samba*. The latter is also *music* as well as a *dance* (Figure 1.10). According to (Freihat et al., 2013), *music-dance* is a *polysemic pattern.*

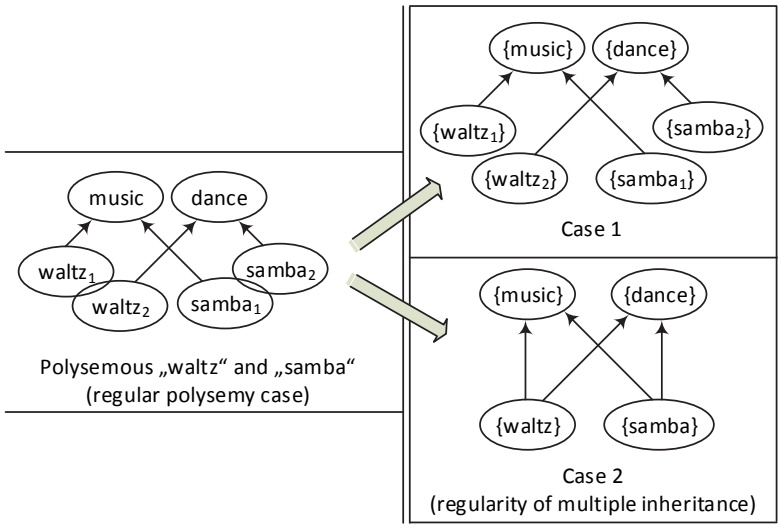


Figure 1.11. *Regular polysemy vs. the regularity of multiple inheritance*

According to (Pethö, 2001), *regular polysemy* usually arises due to metonymy, and *non-regular polysemy* often arises due to metaphors (Figure 1.8 and Figure 1.10).

[30] The idea of the homonymy-polysemy-monosemy axis comes from (Pethö, 2001)

Similarly, the *regularity of multiple inheritance* depends on what method of locating the polysemous words is employed in the wordnet hierarchy (Figure 1.9 and Figure 1.11).

The *regularity of multiple inheritance* means that in the wordnet hierarchical structure there are at least two *synsets* with a minimum of two identical parents (Figure 1.11, Case 2).

1.2.9 Sense clusters of polysemous words

In addition to the fact that polysemous words produce *multiple inheritance* cases in wordnet hierarchical structure, they form certain patterns as well. These patterns, called *sense clusters of polysemous words* (Lin et al., 2002), arise due to the terms of the related meanings are located nearby in the hierarchy. The authors of (Lin et al., 2002) distinguish five types of polysemy patterns for verbs – *sister, twins, child, chain* and *triangle*. (Mihalcea and Moldovan, 2001) refer to these patterns as *a similarity measure* and (Peters et al., 1998) calls it the *clustering method*.

Next, *sense clusters of polysemous words* (polysemic patterns) are described and examples given based on Princeton WordNet (version 3.1). The author of this thesis extracted all the examples.

- **Sisters** are *synsets* that have at least one common word form, and they both have immediate *hyponyms* of the same parents in the wordnet hierarchy (Miller, 1998), (Peters et al., 1998). Based on the verb analysis of (Lin et al., 2002) in Princeton WordNet (version 1.7), *sisters* is the most frequent polysemic pattern of all (Figure 1.12).

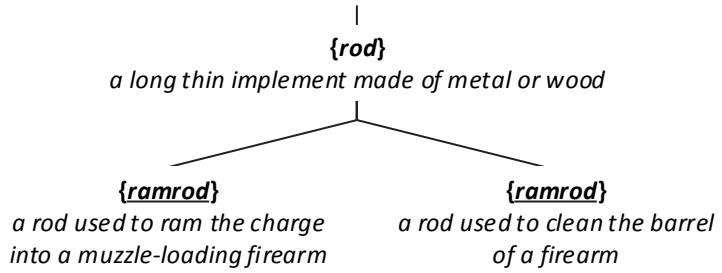


Figure 1.12. *Polysemic pattern – sisters*

- **Twins.** (Miller, 1998) defines twins as *synsets* that have three or more words (*lexical units*) in common. Meanwhile, (Lin et al., 2002) define twins as *synsets* with identical members and use examples where *synsets* have two members (*lexical units*). Despite that, neither of the authors mentions the necessity of a common superordinate; nonetheless, they use the common ancestor of twins (Figure 1.13).

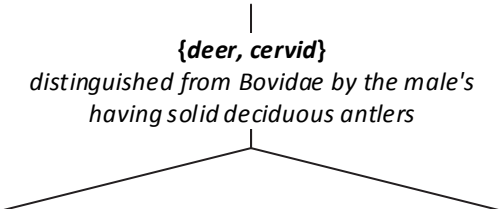


Figure 1.13. *Polysemic pattern – twins*

- **Child.** The same word form exists in a *synset* and its superordinate. Naturally, the subordinate (according to Figure 1.14 {turn}) has a more precise meaning than the superordinate.

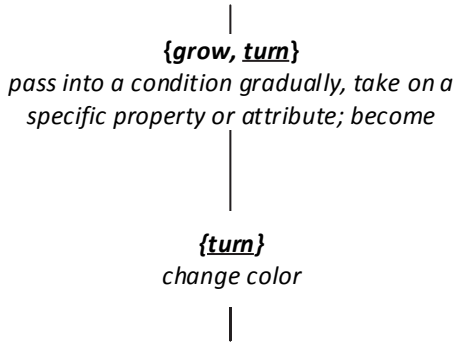


Figure 1.14. *Polysemic pattern – child*

- **Chain.** The same word form (*lexical unit*) appears sequentially in *hypernymic/hyponymic* or *hypernymic/troponymic* chain three or more times. The verb analysis of (Lin et al., 2002) showed that this kind of polysemic pattern is the rarest one (Figure 1.15).

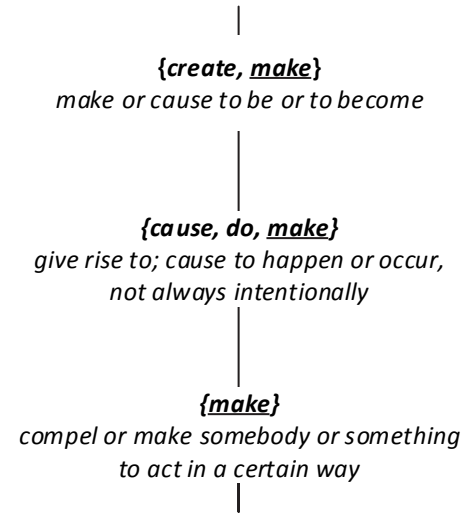


Figure 1.15. *Polysemic pattern – chain*

- **Triangle.** The second rare pattern is a triangle. Here, the sister sense has a *co-hypernym*, which shares the same word form (*lexical unit*) as sisters (Lin et al., 2002) (Figure 1.16).

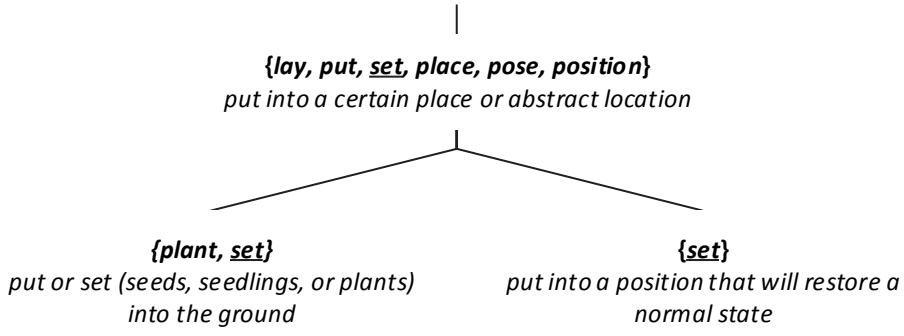


Figure 1.16. *Polysemic pattern – triangle*

1.3 BUILDING A WORDNET

"There are diverse methods of wordnet construction ..."
Maciej Piasecki • Stan Szpakowicz • Christiane Fellbaum • Bolette Sandford Pedersen

The primary structure of a particular kind of lexicon design of every wordnet is based on the "mother wordnet" i.e. on Princeton WordNet (Fellbaum, 1998b). Building a wordnet is a time- and human resource consuming process (Mititelu, 2006), (Sagot and Fišer, 2011). Therefore, developers evaluate the options for wordnet building and choose the most optimal one. There are typically three main aspects in the

building process of wordnet that entail a decision – *what kind of lexical resources to use*; *what kind of building model to use* and technically, *how automated is the building process?* (Figure 1.17).

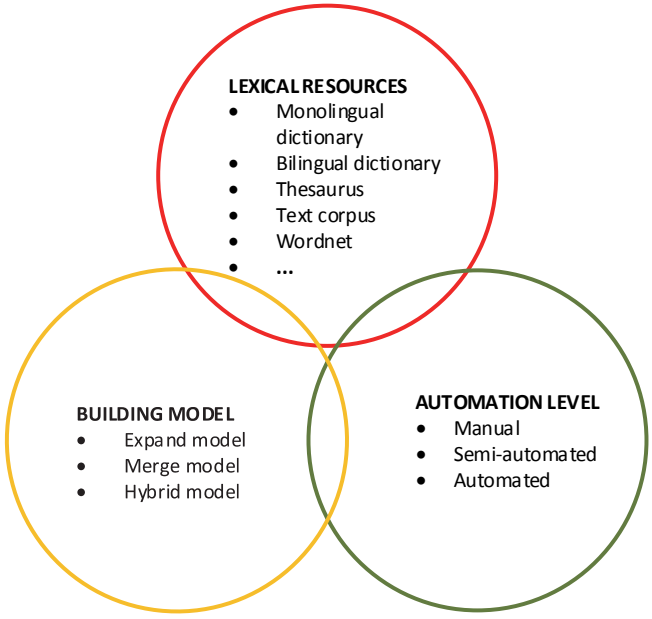


Figure 1.17. *The three building aspects of wordnet building*

In the following sections, these three questions are answered.

1.3.1 Lexical resource

Lexical resources may differ in the various building phases. That is to say, they are not in use only for the first building phase of wordnet but also for the validation/evaluation and the extension process. Very often, developers use different types of lexical resources concurrently. For example, the creators of Czech Wordnet used eight different lexical resources (Pala and Smrž, 2004). At the moment, these resources are contemplated:

- monolingual (explanatory) dictionaries (Prószéky and Miháltz, 2002)(Nadig et al., 2008)
- bilingual dictionaries (Lee et al., 2000) (Thoongsup et al., 2009), (Sagot and Fišer, 2011)
- text corpora (Sinopalnikova, 2004)
- parallel text corpora (Dyvik, 2004)
- comparable text corpora (Kaji and Watanabe, 2006)
- thesauruses (Sinopalnikova, 2004), (Sagot and Fišer, 2011)
- wordnets (Farreres et al., 1998), (Lindén and Niemi, 2014)

- web (Sang, 2007)
- on-line encyclopaedias (Sagot and Fišer, 2011), (Ruiz-Casado et al., 2005a)

Next, every lexical resource is briefly described in order to understand their role in wordnet development. It is important to note that to get beneficial information from the lexical resources, developers use different methods. Together, these methods are usually described as the usage of lexical resources. Furthermore, every resource is applicable to the *building*, *extending* or *validating* processes.

A monolingual (explanatory) dictionary is a lexical resource from where to extract the taxonomical relations like *hypernyms* (Farreres et al., 1998). However, it also provides information about words, i.e. definitions, *synonyms*, domain, usage examples, different meanings, sub-meanings (Langemets, 2010), (Fellbaum, 1998a).

A bilingual dictionary, parallel text corpora and **comparable text corpora** are usually resources for translating *synsets* of the target language wordnet to the source wordnet (usually Princeton WordNet) (Farreres et al., 1998).

Text corpora and **web** are helpful for extracting the semantics relations between the words using the lexico-semantic pattern, as it was typically explained in (Hearst, 1992) and (Snow et al., 2004) (see also Section 2.1.3).

Wordnet is primarily employed as a source for the target language wordnet, translating either its *synsets* (Lindén and Niemi, 2014) or glosses (Kaji and Watanabe, 2006), (Saveski and Trajkovski, 2010).

On-line encyclopaedia such as Wikipedia can be used as a bilingual resource for translating (Sagot and Fišer, 2011) or a monolingual resource for extending the wordnet (Ruiz-Casado et al., 2005b).

Four lexical resources in the list above perform the translating role. However, translating engines could be used instead. For example, (Saveski and Trajkovski, 2010) used Google Translate for translating the glosses of Princeton WordNet for Macedonian WordNet. To clarify, they translated the *synsets* with English-Macedonian MRD (machine-readable dictionary) and the glosses with Google Translate. On both translation (*synsets* and glosses), they applied Google Similarity Distance algorithm (Cilibrasi and Vitanyi, 2007) to choose suitable word candidates for every *synset*.

1.3.2 Building model

Wordnet researchers have introduced two common building models (Vossen, 1998b):

- Expand model
- Merge model

In different literature, these categories are also called the *expansion* and *merge approach* (Prabhu et al., 2012), (Bhattacharyya, 2010).

The expand model takes the source wordnet and translates all of its *synsets* (Lindén and Niemi, 2014), "core" wordnet[31], base concepts (Vossen et al., 1998) or "universal concepts" (Bhattacharyya, 2010). It also takes over all relationships and later expands it with *synsets* from local lexical resources.

The merge model[32] defines the *synsets* and semantic relations in the target language by using existing lexical resources (of target language) such as thesauruses, dictionaries or special text corpora. Then it aligns that wordnet with a "mother" wordnet (e.g. Princeton WordNet or Hindi (Prabhu et al., 2012)) through equivalence relations (Pedersen et al., 2009), (Piasecki et al., 2009).

Advantages and disadvantages of both models

The positive side of the *expand model* is that it saves a lot of time because of the more fluent wordnet constructing process – the lexicographer does not need to think of the concepts for the target language. Secondly, semantic relations can be borrowed from the source wordnet. Thirdly, it "*guarantees the highest degree of compatibility across different wordnets*" (Aliabadi et al., 2014). The negative side of this method arises if the lexicographer is faced with the problems when there are no equivalent concepts for the target language. That is to say, "*the source language may not reflect the richness of the target language*." (Prabhu et al., 2012). Issues of this kind typically occur due to culture and region specific concepts in the source wordnet (Bhattacharyya, 2010). These methods are reasonable if there is a semantic closeness between the source and target wordnets (Farreres et al., 1998). However, not all of the wordnet developers take it into consideration (Kaji and Watanabe, 2006), (Lindén and Niemi, 2014).

As regards the *merge model*, there is no "*distracting influence of another language*". Nevertheless, comparing the *merge approach* to the *expand approach*, the creators have an additional task of finding out, depending on building approach, the *base concepts* or *universal concepts* (Bhattacharyya, 2010). Here, the *base concepts* signify the most frequently represented concepts in the target language (usually in text corpora). The *universal concepts* denote the common concepts across many languages. The *merge model* is a relatively slow one but the expand model is a rather quick one (Bhattacharyya, 2010).

To diminish the shortcomings (time and quality) of both approaches, developers have used both models together, resulting in the **hybrid model** (Prabhu et al., 2012), (Borin and Forsberg, 2014). The time gain is due to the existing lexical structure in the *expand model* and quality is the result of the culture-specific words in the *merge model*.

---

[31] "Core" word senses in Princeton WordNet (approximately the 5000 most frequently used word senses) downloadable from http://wordnetcode.princeton.edu/standoff-files/core-wordnet.txt

[32] It is the dominant model in building BalkaNet (Tufis et al., 2004a) and EuroWordNet (Vossen, 2004)

1.3.3 Automation level

From a technical point of view, the wordnet building process can have three different automation levels: *manual*, *semi-automated* and *automated* (Figure 1.17). These levels can be employed in wordnet creation as well as the extending and validating processes.

Manual wordnet construction is the most reliable and produces the best results because it considers the linguistic soundness and accuracy. However, this approach has two essential drawbacks – *human resource intensive* and *time-consuming* (Fišer and Sagot, 2008). This factor on the one hand and "*the success of recycling already existing language resources, such as bilingual dictionaries, Wikipedia, and parallel corpora*" on the other are the reasons why "*in the past years, automatic creation of wordnets for new languages has become increasingly popular*" (Sagot and Fišer, 2012). Despite the fact that the automatic approach has a significantly lower resource consumption capacity, according to (Nadig et al., 2008) it produces:

a) *synsets* with outlier or missing words,

b) semantic relations that "*may be inappropriately set up or missing altogether*".

Certainly, **every wordnet needs verifying** (Chagnaa et al., 2007), (Sagot and Fišer, 2012) **and it does not matter on which automation level it is composed**. Semi-automatically and automatically built wordnets consume less time and human resources in the wordnet building process, whilst validation is usually not so expensive but still depends on the available resources. Thus, if one assumes that the validation process takes approximately the same amount of time at different automation levels, then the gains in time and human resource are obtained with the semi-automated and automated approach.

Next, some examples are provided for every approach.

Manual approach

An example of a manually composed wordnet is *Finnish Wordnet* (FinnWordNet). Two professionals translated all Princeton WordNet *synsets* into Finnish. Semantic relations were taken over automatically (Lindén and Niemi, 2014).

Semi-automatic approach

In the case of Turkish wordnet, Base Concepts were manually translated from EuroWordNet (Vossen et al., 1998). Subsequently, *synonyms*, *hypernyms* and *antonyms* were automatically extracted from different lexical resources (Bilgin et al., 2004).

Automatic approach

The authors of Macedonian WordNet (Saveski and Trajkovski, 2010) automatically translated Princeton WordNet *synsets* with Macedonian-English MRD and glosses with Google Translate. To find the suitable words for every *synset*, Google Similarity Distance algorithm was used (Cilibrasi and Vitanyi, 2007).

For Persian WordNet (Montazery and Faili, 2010), the authors used Persian and English corpora as well as a bilingual dictionary in order to associate Princeton WordNet *synsets* with Persian words. They calculated "*a score for each candidate synset of a given Persian word and each of its translation*". Based on the maximum score, they selected Persian words to be linked to certain *synset*.

1.4 CONCLUSIONS

Even though this chapter provided some background information for understanding the nature of wordnet and the topics related to the construction of the semantic hierarchies of wordnet. It also gave some insight into how errors may be imported into the semantic network of wordnet.

Expanding on (Piasecki et al., 2013c) "*There are diverse methods of wordnet construction …*" we may assume that similarly there will be many different ways how errors come into wordnet. Undoubtedly, the number of errors in the semantic network of wordnet is directly connected to the approaches the developers decide to use in its construction, expansion and even during validation.

In this chapter, we introduced, among other things, different principles that have to be considered when building a wordnet. We described the principles of *synset* composition, and semantic relations composition. We also introduced principles for how to deal with polysemy, regular polysemy, unique beginners, and top hierarchy in wordnet. Whatever the sources of error are, ignoring these principles entail errors in the semantic network of wordnet.

Sources of errors may vary widely. As wordnet is a human-machine system then of course some errors are introduced by human activities, such as translating concepts (*synset*) from a source wordnet to a target wordnet incorrectly or adding new concepts into a wordnet semantic network and forgetting to link it to other concepts. In addition, human impact may manifest in the different language perception of different lexicographers, and also in changing language perception over time (Čapek, 2012).

Furthermore, the choice of which lexical resources to use in building a wordnet has a strong impact on it. If developers decide to use different lexical resources of a target language simultaneously they have to bear in mind that sense distinctions may vary widely across lexical resources (Peters et al., 1998). When translating a source wordnet to a target wordnet, different languages typically have different semantic spaces and when these languages are not from the same family then culture and region specific concepts also play an important role.

As important as the lexical resources themselves are the methods used for information extraction from them and how automated the process of building and expanding is.

In conclusion, the choice of lexical resources, building model and automation level used in the building or expanding of a wordnet may have a big impact on importing errors into this semantic network.

2. STATE OF THE ART IN VALIDATING THE SEMANTIC HIERARCHIES OF WORDNET

"When building large-scale lexical/semantic resources, subsequent – or better, simultaneous – validation of content is essential" – Dietrich H. Fischer

"Maintaining content integrity and high quality of data in a general purpose semantic network that is in development is of utmost importance for majority of NLP applications in which a wordnet is used" – Tomáš Čapek

In the previous chapter, many rules were discussed that should be considered when building, validating or expanding wordnet. These various rules (regarding *synset members*, *semantic relations*, *polysemy*, *regular polysemy*, *unique beginners, and top hierarchy*) make wordnet a very multidimensional system. On the one hand, ignoring these rules has a big potential to produce errors in the semantic hierarchies of wordnet. On the other hand, quite a few errors might be imported into wordnet hierarchies through the three building aspects considered in Section 1.3 – the *lexical resource*, the *building model* and the *automatization level.* In light of all these rules and building aspects, it is certainly reasonable to check and validate the semantic hierarchies of wordnet. Moreover, this is confirmed by the fact that there are already many methods for wordnet validation.

This chapter studies methods used to check and validate a semantic hierarchy of wordnet. Thus, answers are provided to the following questions: *how to validate the semantic hierarhies of wordnet, what methods find application in the validation of wordnet hierarchies, what features are used to classify them and which group is suitable to our approach* for detecting inconsistencies in the wordnet hierarchical structure. In brief, three groups of methods used in the validation of wordnet are described:

- I group of methods based on lexical resources (Section 2.1)
- II group of methods that use different rule systems to check and validate wordnet relations (Section 2.2)
- III group of methods that utilize particular pattern extraction in the wordnet hierarchical structure as a graph (Section 2.3)

Only two features distinguish these methods – *whether they make use of the lexical resource* and *whether they use the content of a synset*. Table 2.1 gives an overview of the group of validating methods characterized by those two features. Our approach belongs to the third group.

Table 2.1 *Features that classify a group of validating methods*

Group of methods	Whether they make use of the lexical resource	Whether they use the content of a *synset*
I group	+	+
II group	–	+
III group	–	–

Secondly, this chapter provides a description of the different error classes along with examples and answers the following questions: *how to classify the errors in a wordnet system* and *which group of errors does our approach detect*. The errors in wordnet are divided into three classes:

- Syntactic errors – related to the source file structure or data presentation in it
- Semantic errors – related to wordnet semantics
- Structural errors – related to wordnet as a graph

The main results of our work are connected to the second class of errors.

The subsequent sections (2.1–2.3) give an overview of the validation methods of wordnet found in the literature. Various approaches as well as the errors they help to detect are discussed. When possible, error statistics are added to the description. Section 2.4 presents error classes and section 2.5 contains a summary.

This chapter is mainly based on unpublished results. Only section 2.3.1 "*A short overview of the patterns in a hierarchical structure*" is published in "Independent Interactive Testing of Interactive Relational Systems" (Lohk and Võhandu, 2014).

2.1 VALIDATION METHODS USING LEXICAL RESOURCES

The most frequently used validation methods of the semantic hierarchies of wordnet are those which rely on lexical resources. Along with lexical resources, a wordnet developer has to know how to extract beneficial information from them. Some of the well-known approaches are as follows:

- Lexico-syntactic patterns
- Similarity measurements
- Mapping and comparing to ontology or wordnet
- Applying wordnet in some NLP task

The lexical resources used in this group of methods are:

- Monolingual text corpora
- Monolingual explanatory dictionaries
- Web as corpus; included News, Wikipedia
- Wordnets
- Ontologies

It is important to note that here knowledge about the content of a *synset* is essential because it is impossible to extract information about a word from the lexical resource if we do not know what the meaning of that word is.

2.1.1 Monolingual text corpus

When using a monolingual text corpus to validate the items of a wordnet semantic hierarchy, it is presumed that semantically close words are close in the text as well. The same presumption applies when extracting lexico-syntactic patterns. However, patterns in the text are not necessary in that case. While *lexical units* in a *synset* are in the base form, it is important that the usable corpus be lemmatized and POS-tagged.

(Sagot and Fišer, 2012) use **monolingual text corpora** to clean the noisy *synsets* of automatically created wordnets such as French WOLF and Slovene sloWNet. Their approach compares the words presented in the same *synsets* by checking whether these words are used in the same paragraphs of large monolingual corpora. More precisely, comparable words (in term of *lexical units*) come from a base *synset* and from its related *synsets* with a unit distance 0, 1 or 2. The semantic relations used in the relation paths are: *hyponymy, instance hyponym, mero portion, mero part, mero member, eng derivative, holo member, holo part, holo portion, hypernym, instance hypernym*. The result of this experiment is a set of (*lexical unit*[33], *synset*) pairs, where each pair is associated with a final score and the *lexical unit* exists in the text corpus. Leaving aside the five-step calculation procedure of finding different scores (*local_score, global_score, synset_global_score*) for a *lexical unit* in *synsets* and paragraphs, the formula of the final score considers how many times a certain *lexical unit* exists in the different *synsets* and paragraphs of a corpus. Both wordnets use different sources of monolingual text corpus, therefore the authors "*defined empirically two separate thresholds for a minimum score under which a (lexical unit, synset) pair is considered as a candidate outlier*".

The authors of (Sagot and Fišer, 2012) discovered that 67% of the proposed outlier candidates are indeed incorrect for WOLF and a 64% for SloWNet. This is an estimated 12% of the overall error rates in the resources of WOLF and 15% in SloWNet.

2.1.2 Monolingual explanatory dictionaries

This approach relies on the assumption that to every entry in the dictionary there is a corresponding *hypernym* or *synonym*. That idea has a twofold benefit. It is possible to use it for finding new lexical (Blondel and Senellart, 2002) or semantic relations (Nikulásdóttir and Whelpton, 2009) as well as to check them (Nadig et al., 2008).

[33] (Sagot and Fišer, 2012) used the concept of "literal" instead of "lexical unit", which is the term of the database

(Nadig et al., 2008) verified the quality of *synsets* from two perspectives: "*Outlier or missing words in synsets*".

For checking the *synset*, *synonymy* within non-singleton *synsets* was verified. They assumed that "*If a word w is present in a synset along with other words* w_1, w_2, ..., w_k, *then there is a dictionary definition of w which refers to one or more of* w_1, w_2, ..., w_k, *and/or more of the words in the hypernymy of the synset*".

For the validation, three groups of rules on noun *synsets* were applied in a fixed order. If the dictionary definition of the word did not contain any of its *synonyms* or *hypernyms*, it was assumed that two *synonyms* may share common words. They also took into account cases of "*partial matches of hypernyms and synonyms*" of a word. The authors applied defined rules to noun *synsets* which have more than one word (*lexical unit*). For complete validation, they achieved about 70% accuracy, i.e. each word was validated in about 70% of all the noun *synsets*. The remaining 30% of *synsets* were intended for manual checking (Nadig et al., 2008). This 30% included about 9% of noun *synsets* where none of the words were validated.

2.1.3 Lexico-syntactic patterns and the lexical resource

According to (Hearst, 1992): *"There are many ways that the structure of a language can indicate the meanings of lexical items, but the difficulty lies in finding of constructions that frequently and reliably indicate the relation of interest."* That is to say, there are detectable syntactic constructions in the text that indicate the meaning of word(s). (Hearst, 1992) calls these syntactic constructions **lexico-syntactic patterns** and uses them to detect semantic relations such as *hyponymy*. However, this approach is not limited to a single semantic relation. For example, (Arnold et al., 2014) find lexico-semantic patterns in *meronymy*, *holonymy*, and *synonymy* relations.

Although Hearst's idea appears to be quite popular[34]. For example, (Rydin, 2002) uses it to create the hierarchical structure of the lexicon. (Oakes, 2005) asserts that these patterns are "*highly effective*" in extracting semantic relations from pharmaceutical news feeds for automatic thesaurus generation. (Panchenko et al., 2012) found twelve additional patterns for *hypernymy* and *synonymy* relations.

According to (Hearst, 1992), these patterns satisfy the following desiderata:

- They occur frequently and "*across text genre boundaries*"
- They (almost) always indicate the relation of interest
- They require little or no pre-annotated text.

[34] According to Google Scholar, paper of Hearst is referred more than 2,700 times. However, the original idea does not belong to Hearst. (Cruse, 1986) and (Lyons, 1977) discussed the patterns in text many years earlier. Hearst was the first one, who applied lexico-syntactic patterns to WordNet.

Hearst incorporated these patterns into WordNet and used them for

- **verifying** words and their *hyponymy* relations and
- **adding** new nouns and their *hyponymy* relations to wordnet (the augmentation of wordnet)

Heast's algorithm for discovering new lexico-semantic patterns:

Step 1: Initially Hearst discovered two patterns by observation, "*looking through text and noticing the patterns and the relationships they indicate*".

Pattern 1: NP_0 *such as* {NP_1, NP_2 ..., *(and | or)*} NP_n
Pattern 2: NP_1 *(,* NP_2*}*{,} or other* NP_0

Where

NP_0 *is a noun phrase in hypernymy meaning and*
$NP_{1..n}$ *are noun phrases in hyponymy meaning*

Then she applied steps from 2 to 5 repeatedly:

Step 2: gather a list of terms to which this relation belongs to

Step 3: find the places in the corpus where these terms are syntactically close and record the environment

Step 4: find the commonalities among these environments and assume that the common ones yield patterns that indicate the relation of interest

Step 5: in the case of a positively identified pattern, use it to gather more instances of the target relation and go to **Step 2**.

An example of a lexico-syntactic pattern in practice

(Snow et al., 2004) designed an extension to Hearst's patterns by training a *hypernymy* classifier on the basis of dependency trees of known *hypernym-hyponym* pairs. The classifier was effective in detecting all of Hearst's patterns, but it also presented four additional patterns.

(Nadig et al., 2008) employed these ten patterns (6 Hearst's patterns + 4 Snow's patterns) to **validate** *hypernym-hyponym* relations in Princeton WordNet (version 2.1) using automatic search queries on Microsoft Live search. In addition, they applied two other rules (which covered 24.03% of the total *synsets* pairs) to validate *hypernym-hyponym* relations. Lexico-syntactic patterns gave the best results (covered 46.84% of total *synsets* pairs). (Nadig et al., 2008) argue that "*the failure to validate a synset pair is not a definitive indicator of erroneous construction and has to be treated as a flag for human inspection*". About 30% of *synsets* pairs required this kind of check.

2.1.4 Applying wordnet in some NLP tasks

It appears that a very effective way to detect the shortcomings in a wordnet semantic network is to employ it in semantic analysis tasks. That will clarify how good the quality of wordnet is, i.e. how sufficient the vocabulary of wordnet is. On the other hand, it should be enquired whether all the senses are distinguishable in wordnet.

Lexico-semantic annotation task

(Hajic et al., 2004) employed the Czech WordNet for the **lexico-semantic annotation** of the Prague Dependency Treebank (Böhmová et al., 2003). They used statistics of the annotated data as feedback in order to validate and improve the coverage and quality of the Czech WordNet. Based on the experience of authors (Hajic et al., 2004) certain issues concerning the coverage and quality of the Czech WordNet were highlighted. They found that:

- Less than 50% of the nouns, adjectives, and verbs in annotated texts appeared in the Czech WordNet.
- With the help of the Czech WordNet, only 30% of the nouns, adjectives, and verbs in annotated texts were successfully annotated.
- The Czech WordNet did not cover some of the very common meanings of frequent words.
- Only 12% of all the *synsets* of the Czech WordNet were assigned to the words in the annotated texts.

These four facts were evidence of a) the uneven distribution of the *synsets* of the Czech WordNet and b) the insufficient word coverage. A Czech WordNet team applied some of this feedback to the validation and improvement of the quality of wordnet, by changing, deleting and adding certain new *synsets* (Hajic et al., 2004).

Word Sense Disambiguation (WSD)

One of the goals when using wordnet in WSD task could be to evaluate its utility (Kahusk and Vider, 2002), but also to help to discover:

- Which words are not represented in wordnet and
- What words cannot be extracted from wordnet

The two cases below provide a more specific illustration.

An example from (Saito et al., 2002)

(Saito et al., 2002) evaluate the adequacy of GermaNet for the WSD task. In other words, they were interested in how useful GermaNet is as a lexical resource for the WSD task.

Their second purpose was to obtain clues for improving GermaNet. For that reason, a small automatically lemmatized and POS-tagged corpus was composed. In addition, a special software tool for that task helped five annotators to select the

suitable senses. They gathered all the cases where the counterpart for the token was represented in GermaNet but not determined by annotators. On the basis of non-determined tokens, error classes were composed. These are presented in a decreasing order by their frequency.

Auxiliary (word) belongs to the verb. Special meaning for this word is not allowed.

Compound. The problem with compound words is that there are infinite possibilities to compose them, especially in German, but GermaNet is not capable of describing all of these.

Lemma. In 15% of the cases, the lemmatizer determined a wrong lemma to a word. It is impossible to choose the right meaning for a word if the basic form is wrong.

Other are the words that could be or should be in GermaNet but are not there.

Derivations are words where the noun is derived from the verb (e.g. Vorbereitung from vorbereiten), and also generations from the diminutive form (e.g. Hündchen from Hund).

Particle significantly changes the meaning of the verb. Sometimes it is concatenated with the verb, e.g. *vorschlagen – "propose"* and sometimes not, e.g. *Er schlug einen Kompromiss vor – "He proposed a compromise"* (Both examples from (Saito et al., 2002)). The latter case "*presents difficulties for lemmatizers*".

Collocation. Many words have specific meanings in combination with other words. This category also includes idioms (e.g. *ins Wasser fallen – "cancelled"*). The authors of (Saito et al., 2002) state: "*While it is arguably not the task of a lexicon to account for collocations and idioms, we were interested in assessing the degree to which these are problematic.*"

As a result of that annotation, on average 92% of the words (nouns, verbs, adjectives) were given at least one sense by GermaNet and more than 83% of the words received at least one sense that was assessed as the correct sense by five annotators.

To summarize, the authors of (Saito et al., 2002) concluded that many types of errors were clearly German-specific. This in turn means that "*language-specific issues are quite important when evaluating the effectiveness of a particular WordNet*". A second important inference by the authors was that it is possible to significantly improve the sense-tagging by GermaNet "*integrating additional morphological processing into the tagger*". Notably, the methods for compound words and derived words could improve the sense tagging significantly.

An example from (Kahusk and Vider, 2002)

(Kahusk and Vider, 2002) applied the WSD task to Estonian Wordnet (EstWN). The primary goal was to assess "*how well the existing EstWN covers real language usage in texts*".

At the time, there was no manually disambiguated text for the Estonian language. The authors of (Kahusk and Vider, 2002) decided to create a reasonable amount of that kind of text. Before the manual disambiguation task, they used the morphological analyzer ESTMORF to find out for every word its senses with lemmas and word classes (part-of-speech). After that, four linguists disambiguated the nouns and verbs in the texts – two linguists for each text. The sense number of the word marked by the linguists followed the sense number in EstWN. If a word was missing in wordnet, a linguist marked its sense with a "0", and if a word was in EstWN but did not have the appropriate sense, it was marked with a "+1".

The authors found that about 46% of the words not represented in wordnet were **compounds**. An indefinite number of compound words in the Estonian language contributes to this problem. It is easy to compose a new compound in the Estonian language that is not found in any dictionary.

Secondly, a noteworthy word category was the **proper name**. EstWN does not contain words from the proper name category. As a result, about 17.5% of such words in analyzed texts are not in EstWN.

If phrasal words and some strange words with hyphens (about 7%) are discarded, the most valuable outcome of the WSD task was uncovering about 29% of words (not represented in wordnet) that are suitable for adding to EstWN.

2.1.5 Comparing wordnet to another wordnet through ILI

There is no doubt that lexical concepts from different languages have a different spread of semantic fields. Therefore, it is not possible to automatically transfer all lexical concepts to another language. Moreover, there are language concepts represented in one language not represented in the other, and also vice versa. For instance, there are four different meanings of the word "eat" in the Thai language according to social status (Thoongsup et al., 2009). In Dutch, there are no words for top-level concepts of a *container* (an object used to hold things) (Fellbaum and Vossen, 2008). In Finnish, there is a lack of "*words for inhabitants of Finnish towns and provinces*" (Lindén and Niemi, 2014).

If these exceptions are discarded, references can be set between the common concepts of different wordnets. The same idea is also followed in EuroWordNet through the use of the **Interlingual Index** (ILI). This is a list of lexicalized concepts which appear in at least one wordnet of the EuroWordNet. Thus, "*ILI entries merely function to connect equivalent words and synsets in different languages. Equivalent relations between the synsets in different languages and Princeton WordNet are made explicit in the ILI.*" (Fellbaum and Vossen, 2008).

Comparing wordnets through ILI helps to evaluate[35] whether the differences between the semantic hierarchies of wordnet are justified or if there is a lack of *concepts*, *synonyms in the synset*, *coverage of concepts* or some other shortcut (Pedersen et al., 2012).

Comparing regular and metonymic polysemy

(Peters et al., 2002) demonstrated in their paper that ILI is useful for particular tasks. They enquired whether the phenomena of *metonymic polysemy* and *regular polysemy* carries across languages. Three wordnets from EuroWordNet were utilized in their work – *English*, *Dutch*, and *Spanish*. As regards *metonymic polysemy*, it was concluded that it is language-specific.

Their manual evaluation shows that a *regular polysemy* pattern is valid across three languages and it has a certain level of universality. The results of the experiments also revealed the "*potential for enhancing the semantic compatibility and consistency of wordnets*". It emerged that on the basis of *regular polysemy*, wordnet can be automatically extended from other wordnets. In their "*small experiment 50% of the Dutch and Spanish words that do not display a WordNet-derived regular polysemic pattern were successfully semantically enriched with this pattern*" (Peters et al., 2002).

META-NORD project

(Pedersen et al., 2012) introduce 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 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) set a goal to link all Nordic and Baltic wordnets to the 5,000 "core synsets" of Princeton Wordnet. These 5,000 most frequently used English word senses are a subset of Princeton WordNet compiled semi-automatically (Kahusk et al., 2012). For resource comparison, four measurements were to be used:

- Taxonomical structure
- Coverage
- Granularity of the described concepts
- Completeness of a *synonym*.

[35] The higher purpose of using the ILI is multilingual processing (Tufis et al., 2004b)

2.2 DIFFERENT RULE SYSTEMS TO CHECK WORDNET RELATIONS

This section discusses a group of methods which do not use any lexical resources as background knowledge bases, but consider the content of lexical relations (word-word), semantic relations (concept-concept) and the rules between them. Naturally, every wordnet developer must have regard to the basic principles of *synset* and semantic relations given in the WordNet "bible" (Fellbaum, 1998a) and described in sections 1.2.2 – 1.2.5. This work adds a few other approaches using:

- Metaproperties of ontology analysis
- Non-expert human annotator in crowdsourcing
- Top Ontology features
- Specific rules for particular error detections

2.2.1 Using metaproperties of ontology

(Guarino and Welty, 2002) propose a methodology to analyze ontologies. The methodology, based on formal notions, is called OntoClean. These notions define a set of meta-properties – *rigidity*, *identity*, *unity* and *dependence* – that "*impose several constraints on the taxonomic structure of an ontology, which help in evaluating the choices made*" (Guarino and Welty, 2002). In this manner, they can *„avoid formal contradiction and unsound inheritance of properties"* (Hicks and Herold, 2011). The OntoClean methodology is applicable to subsumption relations such as *hyponymy*. The authors of (Gangemi et al., 2002b) applied the OntoClean methodology to the Top-Level taxonomy of Princeton WordNet and promised a taxonomy that "*is meant to be conceptually more rigorous, cognitively transparent, and efficiently exploitable in several applications*".

The focus here is on the **meta-property of *rigidity***, as it is a most easily discoverable pattern. Rigidity is based on the philosophical notion of essence. "*A rigid concept is a concept that is essential to all of its possible instances.*" *„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* relations in Princeton WordNet may sometimes be a *role* or *type* relationship. The authors of (Lohk and Võhandu, 2014) refer to cases where rigidity checking was employed in certain *hyponymy* relations. The main idea is that if a super concept (*synset*) is a rigid concept then the semantic relation should be *role*, but when the super concept is a non-rigid concept then the semantic relation should be *type*. In order to check the relations of *type* and *role* one can ask:

1. *Is X always or necessarily a Y?*
2. *Can X stop being a Y?*

If the answer to the first question is "yes" or to the second one "no", then the semantic relation should be *type*, but in the opposite case it should be *role*.

For example, let us use the example of (Hicks and Herold, 2011) of *animal-cat-pet* and let us check which kind of relation there should be. A cat, being an animal is an essential concept, because it is impossible for a cat not to be an animal. A pet is a non-rigid concept since not all cats are pets.
Thus, the relations between *animal – cat – pet* have to be as follows:

Animal – (type) → cat – (role) → pet

Another important way to check the correctness of the hierarchy by using the rigidity property is to follow the idea that ***roles* cannot subsume *types***. Therefore, if we have a sequence of *animal – pet – cat*, then according to the previous example, this sequence is wrong:

Animal → pet (non-rigid/role) → cat (rigid/type)

OntoClean is not equipped with methods for determining the meta-properties of a given concept within an ontology. That is to say, all the annotation takes place manually, which is a time-consuming process and in turn, a very expensive task. In addition, the level of agreement among the human annotators is low. In light of this, there are tools (*Rudify, AEON*) for automatic OntoClean meta-properties detection (Hicks and Herold, 2011), (Völker et al., 2005).

OntoClean is integrated into different ontology editors such as OntoEdit (Sure et al., 2003), Protégé (Noy, 2003).

In (Oltramari et al., 2002), the authors apply the OntoClean approach to restructuring a WordNet's top-level. As a result, the wordnet is promised to be "*more rigorous, cognitively transparent and efficiently exploitable in several applications*".

2.2.2 Crowdsourcing

Based on the Merriam-Webster dictionary[36], crowdsourcing is "*the practice of obtaining needed services, ideas, or content by soliciting contributions from a large group of people and especially from the online community rather than from traditional employees or suppliers*".

(Lindén and Niemi, 2014) employed the crowdsourcing procedure for **evaluating synonyms** of **Finnish wordnet** (FinnWordNet) translated from Princeton WordNet by professional translators. The users were assigned to evaluate the quality of the sets of *synonyms* on the scale of 1–5 in the context of an English gloss, part-of-speech and *hypernym* in the FinnWordNet web search interface. Over about two years, users submitted ratings for 1,237 *synonyms*. Manual examination and, if required, correction was applied to *synonyms* with a poor grade. *Synonyms* received a poor grade (1 or 2) 317 times, which is about 25.6% of all grades.

[36] http://www.merriam-webster.com/dictionary/crowdsourcing

2.2.3 Top-Ontology features

(Atserias et al., 2005) introduce the work they "*carried out towards the so-called shallow ontologization of WordNet*". The aim was to overcome most of the many structural problems of WordNet classifications. They concentrated on the following structural problems "*since they violate the nature of the IS-A relationship*":

- "*There is no distinction between instances and categories*"
- "*Some specializations (hyponyms) contradict their categories' (hypernyms) nature*"
- *"Exclusivity between categories is not always clear (unclear multiple inheritance)"*

The authors of (Atserias et al., 2005) utilized Top-Ontology (TO) (Rodríguez et al., 1998) as concept features which allows to find *synsets* that bear "*contradictory information*" because "*category disjunctions and incompatibilities are explicitly declared in the TO*". In their examples, TO nodes were used as concept features of *Object*, *Substance*, *Plant*, *Comestible* and *LanguageRepresentation*.

The top of the taxonomy of *body_covering_1* was studied, where *Object* (countable) and *Substance* (uncountable) were underspecified. They found many conflicts where *synset* members or *co-hyponyms* had contradicting features. For example, in a *synset* of {*plumage_1, feather_1*}, *plumage_1* is uncountable and *feather_1* is countable. As regards *co-hypernyms* {*skin_4, pelt_2*} and {*skin_1, tegument_1*}, "skin" is an *Object* in the former and a *Substance* in the latter.

For corrections, they used the idea of blocking inheritance in those edges where subsumption errors appear and linking it to a basic TO.

2.2.4 Specific rules for particular error detections

Here, some examples from different authors are presented.

(Gupta, 2002) utilizes formal consistency checks for the subsumption (*hyponymy*) relation. Two different queries were applied to uncover *hyponymy* relations, which do not meet the requirements. He presumed that if one of the queries gives a positive result, then the *hyponymy* relations on these points are wrong. The two queries provided answers to the following questions:

- *"Are there opposed concepts where one subsumes the other?"*
- *"Are there opposed concepts which have a common subconcept?"*

It is noted that two concepts are opposed (or synonymously 'antosemous') if at least two of their *lexical units* are antonyms.

Prefix forms as an indicator of hypernymy

The authors of (Nadig et al., 2008) present the relationship between *synsets* where the member of one *synset* is used as a suffix for the member of another *synset*. They utilize examples like {work}, {paperwork}, and {racing}, {auto racing, car racing}.

"If one term of a synset X is a proper suffix of a term in a synset Y, X is a hypernym of Y"

(Nadig et al., 2008) tested Princeton WordNet (version 2.1) and found that 21.35% of the relations corresponded with the abovementioned rule. All these relationships need human inspection.

2.3 PATTERNS IN A HIERARCHICAL STRUCTURE

This group of methods can be described as the most formal one and it does not take into account the semantics of *synsets* and relations. This group contains items such as *cycles*, *rings*, *dangling uplinks*, *orphan nodes*, *small hierarchy* and *unique beginners*. In the majority of cases, the authors of different papers present these patterns as suggestions for checking the quality of the wordnet hierarchical structure. Therefore, a brief overview of them is given. In addition, two query languages are introduced, which have created especially wordnet-like lexical databases.

2.3.1 A short overview of the patterns in a hierarchical structure

Cycles. Despite the fact that a cycle seldom appears in wordnet, many authors have put forth the *cycle* as a test for checking wordnet accuracy (Šmrz, 2004), (Kubis, 2012).

Rings are subgraphs where a node has at least two parents, which in turn have common ancestor (Fischer, 1997), (Liu et al., 2004), (Richens, 2008), (Section 3.1.2). Figure 2.18 represents two artificial examples of rings. (Liu et al., 2004) utilizing Princeton WordNet (version 2.0) found rings within semantic categories and part-of-speech. As a result, she detected 1,837 rings from the noun hierarchy and 17 rings from the verb hierarchy.

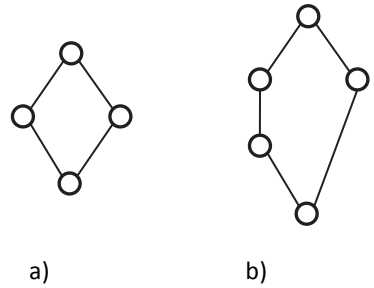


Figure 2.18. *Rings, two artificial examples*

Dangling uplinks are subgraphs where a node has two parents, one connected to a "big hierarchy" and the other to parents, which do not have any superordinate or additional subordinates (Koeva et al., 2004), (Šmrz, 2004).

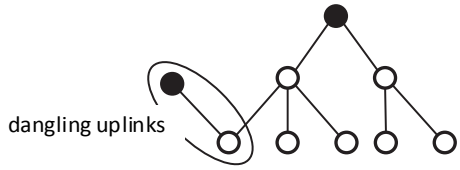


Figure 2.19. *Dangling uplinks, an artificial example (filled nodes represent root synsets)*

Orphan nodes (or null graphs) in the context of a wordnet semantic network are *synsets* without any semantic relations or *synsets* without *hyponymy/hypernymy* relations (Čapek, 2012).

Small hierarchies are subgraphs which end after the root of the nodes on the next three levels (Lohk et al., 2014c).

Root nodes in the context of a wordnet semantic network are unique beginners. The primary need for their discovery is to get an overview of all of them (Lohk et al., 2014c).

2.3.2 Query languages in hierarchy checking

It is possible to apply a query language (MySQL, MSSQL, PostgreSQL, and others) to a wordnet hierarchical structure if it is stored in the database. This approach presumes that the user is aware of the possible points of inconsistency. For instance, in such circumstances it is quite easy to uncover the following cases using query languages:

- All the *synsets* without any semantic relations (orphan nodes)
- All the *synsets* without any parents (top/root nodes)
- All the *lexical units* not related to any *synsets*
- All the *synset* pairs without an opposite semantic relation (i.e. for a *hyponymy* relation, the opposite relationship is *hypernym*; a database should contain both of them)

One of the most capable query languages for wordnet-like lexical databases is **WQuery** (Kubis, 2012). WQuery creators show that one of its roles is to obtain information about a wordnet hierarchy (Vetulani et al., 2010). The WQuery system can operate on wordnet-related terms like *synsets*, word senses and words (Vetulani et al., 2010). Author of WQuery advertises its language as one that is more capable than other wordnet development query languages, such as **WN** (Koeva et al., 2004) and **Hydra** (Rizov, 2008) which do not encompass arithmetic expressions and are not able to answer aggregate queries. WQuery is used as a supporting tool in the development of **PolNet** (Polish WordNet) and it "*is particularly useful to* ***deal with complex querying tasks*** *like* ***searching for cycles in semantic relations***, ***finding isolated synsets*** *or computing overall statistics*." (Vetulani et al., 2010). WQuery can carry out *tree queries*,

reachability queries as well as queries for finding the least common subsumers. The queries of WQuery are also able to:

- Compute structural measures such as minimum/maximum depth and height of the *hypernymy* hierarchy (*tree query*)
- Uncover all the paths that connect two given *synsets* through *hypernymy* (*reachability query)*
- Cycles in a *hypernymy* relation
- *Synsets* that do not entail a *hypernymy* relation
- The number of top *synsets*

An advantage of using of WQuery is its module WUpdate that enables to import any wordnet that is stored as the Global WordNet Grid format in an XML document.

2.4 CLASSIFICATION OF WORDNET ERRORS

Even though different error classification means for wordnet have been proposed by different authors (Koeva et al., 2004), (Čapek, 2012), this work chiefly follows the idea of (Piasecki et al., 2013a) but instead of formal errors, the notion of syntactic errors is utilized as in (Tengi, 1998). For classification, all wordnet errors are divided into three different levels or classes:

- Syntactic errors – related to the source file structure or data presentation in it
- Semantic errors – related to wordnet semantics
- Structural errors – related to wordnet as graph

Descriptions of all of these error classes are provided with ample examples. As in literature there exist several other error classifications, their appropriateness in our system is also considered.

2.4.1 Syntactic errors

Syntactic errors, or *formal errors* (Piasecki et al., 2013a), are those that appear in wordnet source files, i.e. primarily XML-files syntax. (Koeva et al., 2004) upon referring to a similar error class use the notion of *surface errors* which are "*directly present in lexical units, synset literals, glosses, or other metadata thereof*".

- Empty ID, POS, SYNONYM, SENSE
- XML tag data types for POS, SENSE, TYPE (of relation), characters from a defined character set in DEF and USAGE
- Duplicate IDs
- Duplicate triplets (POS, literal, sense)
- Duplicate literals (*lexical units*) in one *synset*
- Typographical errors
- Spelling errors

- Incorrectly entered words

(Šmrz, 2004), (Koeva et al., 2004), (Čapek, 2012), (Lindén and Niemi, 2014).

2.4.2 Semantic errors

Semantic errors are those which are connected to the semantic of *synsets* and relationships. *Taxonomic inconsistencies* (Alvez et al., 2008b) also belong in this class. While not all examples originate from wordnet developers we do not claim that every item in the following list points to a semantic error but rather to a **possible semantic error**.

- Wrong or missing semantic relation
- Wrong or missing *synset* – "*some sets of words used as synonyms, e. g. {"slump"; "crash"; "bust"} are not encoded as synonyms in WordNet*". (Richens, 2011)
- Inappropriate *lexical unit* in a *synset* – {plumage_1, feather_1} in Princeton WordNet (version 3.0), plumage is uncountable, feather is countable (Atserias et al., 2005)
- *Synset* with wrong gloss (definition), the included definition is equal to the *lexical unit* (the latter was a problem in Estonian Wordnet version 70, where 118 *synsets* corresponded to this criteria[37])
- Malapropism – "*the confounding of an intended word with another word of similar sound or similar spelling that has a quite different and malapropos meaning. For instance, an ingenuous [for ingenious] machine for peeling oranges*" (Hirst and St-Onge, 1998).
- Not justified or "unfinished" *multiple inheritance* – based on Estonian Wordnet (version 64) {hote1_1, ...} and {hostel_1, ...} had two parents {institution} and {building}, but {motel_1} had only parent of {institution} (Lohk et al., 2014a)
- Polysemy consistency – the *synset* {letter_1} (a written message addressed to a person or organization) in Princeton WordNet (version 3.0) inherits both its abstract content from its *hypernym* {text_1} and its physical aspect from its *hypernym* {document_2}. Meanwhile the *synset* of {book_1} (a written work or composition that has been published (printed on pages bound together)), a rather similar case, "*is not accounted for in this way*" inherits only physical aspect (Verdezoto and Vieu, 2011)
- Reduction of sense – "*a reduction of sense occurs whenever a hypernym accounts for a part of the meaning of one of its hyponyms*"; {counterfoil_1, stub_4} is a part of a check that provides information about a money transfer. Following the line of an "*inherited hypernym*", it goes up to {abstraction_1}. No single ancestor takes into account the fact that a counterfoil is a piece of paper.

[37] Not yet published experiment of thesis author

Instead, its ancestors refer to the information the counterfoil carries (Alvez et al., 2008a).

- Overgeneralization – no subordinate of {social group} does not include population and generation. "*The nearest hypernym of {social group} that does include population and generation is {group}, but this is an overgeneralization as it subsumes groups of non-living things as well*" (Měchura, 2010).

2.4.3 Structural errors

According to (Piasecki et al., 2013a), structural errors are those "*that can be identified on the basis of the relation definitions and the link structure without going more deeply in the semantics of the wordnet elements linked (i.e. synsets or lexical units)*". This class includes all the errors that are described in Section 2.3 *Patterns in a hierarchical* structure.

2.5 CONCLUSIONS

This chapter presented a variety of approaches for checking the condition of a wordnet. It was shown that based on two features (*whether it uses lexical resources* and *whether it uses the content of synset*), it is possible to categorize all these approaches into three groups of methods.

It appears that the biggest group of methods is the first one, which uses lexical sources as well as the content of *synsets* (i.e. considering semantics). The large number of approaches is due to the fact that there are a lot of machine-readable lexical resources (of course, not for every language version of wordnet) and several approaches for extracting beneficial information from them. For instance, to obtain beneficial information from a text corpus, three approaches are introduced – lexico syntactic patterns (Section 2.1.3); NLP tasks (Section 2.1.4); calculating different scores between *lexical units* and *synsets* or paragraph (Section 2.1.1).

The smallest group of methods (Section 2.3) is the third one since it uses neither lexical resources nor the content of *synsets*. These methods are based on different pattern extractions in the wordnet hierarchical structure as a graph. All test patterns introduced in the next chapter belong to the third group.

It is worth noting that sometimes when enhancing wordnet quality, approaches from different method groups find application. For example, (Mihalcea, 2003) employed two groups of methods for reducing polysemy in Princeton WordNet. The first method was based on polysemy patterns in wordnet and did not use additional lexical resources, and second one relied on the sense tagged corpus SemCor[38].

The second big issue considered in this chapter was error classifications along with their descriptions (Section 2.4). The idea of three error classes of a wordnet system

[38] http://globalwordnet.org/wordnet-annotated-corpora/

proposed by (Piasecki et al., 2013a) was discussed – *syntactic errors, semantic errors,* and *structural errors*.

(Piasecki et al., 2013a) intimated that syntactic errors "*do not occur when a system like WordnetLoom* (Piasecki et al., 2013b) *is used for wordnet development*". It is obvious that these errors are not considered to be particularly important (looking at their statistics) because every development system should be capable of guaranteeing the quality of the syntactic data. Yet, the most significant errors in a wordnet are the semantic errors because a wordnet is first and foremost a semantic network. The significance of semantic errors is reflected in about 20 of the proposed approaches regarding the three groups of methods, both directly (in first and second group) and indirectly (in third group[39]).

In next chapter, we complement *patterns in a hierarchical structure* with our *test patterns*, which belong to the third group of methods.

[39] Every structural error is also caused by a semantic error

3. TEST PATTERNS

"It should always be quicker to implement a test, if we can find a pattern in the data, rather than to do a full revision in top-down or alphabetical order." – Tomáš Čapek

"Structural errors are harder to find and sometimes hard to define. They deal with correctness and appropriateness of lexical and semantic relations among synsets ..." – Tomáš Čapek

The aim of this chapter is to introduce the test patterns that form the basis for the evaluation of a wordnet hierarchical structure and its semantics.

Test patterns, by their nature, are descriptions of substructures with a specific nature in the wordnet semantic hierarchy as a graph. In this work, the focus is on substructures that have the property of *multiple inheritance*. In the most cases, behind the *multiple inheritance* is a polysemy (Sections 1.2.7–1.2.8), but in the remaining cases, there are nodes (*synsets*) that inherit simultaneously specific and general concepts (Section 3.2.1).

Test patterns' structures overlap each other partially or entirely. However, they have different perspectives to the substructures of hierarchies and may typically point to different semantic errors in these.

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

Motivation

There are many reasons why test patterns should be chosen as a way to check and validate *multiple inheritance* in the wordnet hierarchical structure (formed by its semantics). To begin with, *multiple inheritance* itself provides many reasons for checking it:

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 patterns 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 every language (Lohk et al., 2014c).

The main question answered in the following sections is *what kind of test pattern to use in multiple inheritance cases.*

What the most similar work to ours is will also be discussed. These are two test patterns termed a *short cut* and a *ring* (Fischer, 1997), (Liu et al., 2004), (Richens, 2008). The first pattern (*short cut)* represents an exception in this set of test patterns. Whilst it occurs in the case of *multiple inheritance* polysemy is not the cause of its existence.

Section 3.3 considers *what kind of errors are typical to every test pattern*. In brief, some errors are typical to some test patterns. However, frequently they may lead to common errors in the semantic structure of wordnet, such as a *missing or redundant semantic relation*, *missing or redundant sense*, *missing or redundant lexical unit*, etc.

This chapter is based mainly on the paper "New Test Patterns to Check the Hierarchical Structure of Wordnets" (Lohk et al., 2014b). It is also partly based on the following papers: "First Steps in Checking and Comparing Princeton WordNet and Estonian Wordnet" (Lohk et al., 2012b), "How to create order in *large closed subsets* of wordnet-type dictionaries" (Lohk et al., 2013), "Independent Interactive Testing of Interactive Relational Systems" (Lohk and Võhandu, 2014), "Some structural tests for Wordnet with results" (Lohk et al., 2014c) and "Dense Components in the Structure of Wordnet" (Lohk et al., 2014a).

3.1 RELATED WORKS

This section studies other authors' *test patterns*, which in their nature are quite similar to our work. It can even be said that our work is complementary to and generalization of the given patterns.

3.1.1 Short cut

According to section "Lexical hierarchy" written by George Miller (1998) it can be inferred that "*redundancy-free data was the aim of WordNet lexicographers*" (Fischer, 1997). Redundancy in a wordnet hierarchical structure expresses itself if two *synsets* have a twofold link between them – the first link is direct and the second one indirect through another *hyponym*(s)-*synset*(s) relation. (Fischer, 1997) refers to this direct link as a *short cut* (in Figure 3.1 "a" and "b" the arrows with no fill) and (Richens, 2008) calls it *asymmetric ring topology*.

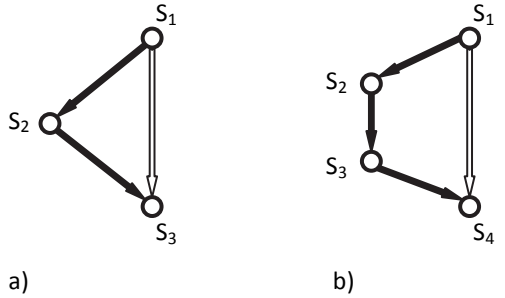


Figure 3.1. *Short cuts in wordnet hierarchy*

In Figure 3.1, there are two images of *short cuts*. Nodes S_1, S_2, S_3, and S_4 denote *synsets* and the arcs between the nodes denote semantic relations. In image (a), the redundant link is between S_1 and S_3. In image (b), the redundant link is between S_1 and S_4. The short cut problem arises because the principle of economy is ignored (Vider, 2001) and it does not originate from polysemy. Hence, *synset* S_3 in image (a) and S_4 in image (b) are not ambiguous concepts.

The *short cut* may occur if lexicographer has created a new, more precise link to another *synset*, and forgot to remove the previous relation.

3.1.2 Ring

According to the approach of (Richens, 2008), a *ring* is a substructure of a wordnet hierarchical structure where one subordinate (in Figure 3.2 (a, b, c), nodes S_5, S_6, S_4) has a superordinate (in Figure 3.2 (a-b), node S_1) via two branches. Richens referring to the work of (Liu et al., 2004), distinguishes two types of *rings*: *an asymmetric ring topology* and *a symmetric ring topology*. In the case of *asymmetric ring topology*, the lengths of both chains in the branches are different in Figure 3.2 (a). As regards *symmetric ring topology*, the lengths of all the chains in the branches are equal in Figure 3.2 (b) and (c). Based on these claims, the length of the chain in different branches of *asymmetric* and *symmetric rings* may be longer than represented in the images in Figure 3.2.

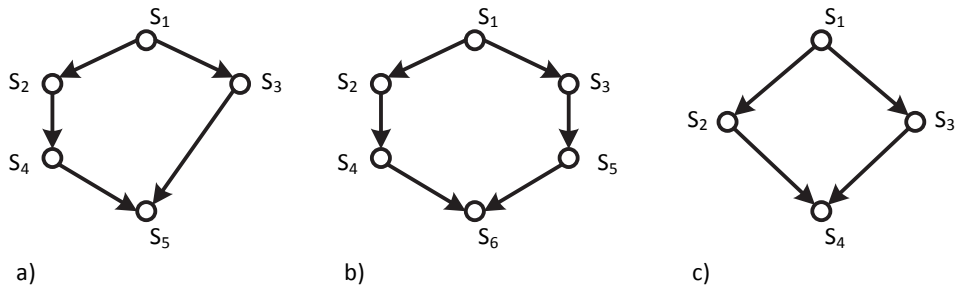


Figure 3.2. *a) an asymmetric ring, b) and c) symmetric rings*

According to (Liu et al., 2004), a ring is unavoidably formed if a *synset* "*has at least two fathers in its own category*". Applying Liu's example to Figure 3.2c", (Richens, 2008) phrases Liu's description of inconsistency in a ring as follows: "*a ring is a paradox because it assumes that two hyponyms [S2 and S4] of a single hypernym [S1] must have opposite properties in some dimension and therefore cannot have a common hyponym [S4], as a hyponym must inherit all the properties of its hypernym[s] [S2 and S4]*".

3.2 NEW TEST PATTERNS

This section is studies test patterns proposed by the author of this thesis. An exception is the test pattern of *synset with many roots*, which instances may in some cases correspond to substructure called *dangling uplink* (Figure 2.19) referred by (Koeva et al., 2004), (Šmrz, 2004).

Filled nodes in the following figures represent root synstes, i.e., synsets without superordinate synsets.

3.2.1 Synset with many roots

(Richens, 2008) and (Liu et al., 2004) introduce the *multiple inheritance* cases which form *rings* (Figure 3.2, a, b, c). However, in addition to the *rings* there also exist *synsets* with many parents, which do not form the *rings*. Instead, their branches flow into different *unique beginners* (*root synsets*). In Figure 3.3, these root nodes are depicted as filled nodes. The benefit of this perspective is an overview of the threaded hierarchies as well as the possibility to evaluate whether the connections between the *synset* with many parents and its roots is justified.

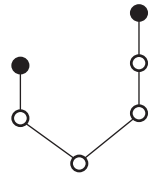


Figure 3.3*. Test pattern of a synset with many root synsets*

3.2.2 Closed subset

Closed subset in our work are coherent bipartite graph in two sequential levels of a wordnet hierarchical structure (Lohk et al., 2013). Figure 3.4 presents an artificially constructed hierarchical structure with one root node (*root synset*). *Closed subset* cases are highlighted by rectangles. We are interested in the cases with at least two parents (represented by thick lines), i.e. where multiple inheritance is used.

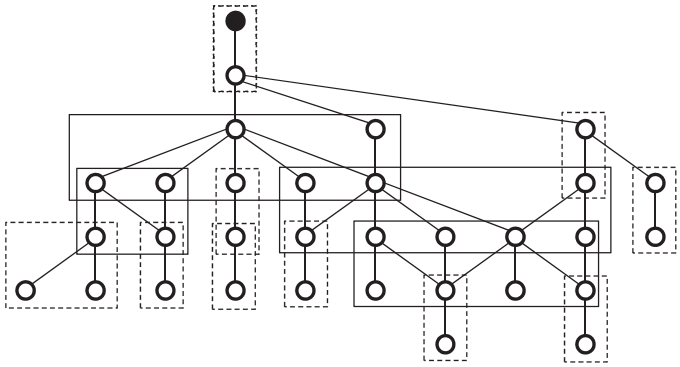


Figure 3.4. *An artificially constructed tree of a wordnet with closed subsets*

The benefit of the *closed subset* is a cluster of tightly connected concepts in particular hierarchical levels (see Figure 3.4). Often it demonstrates connections between two grouped subconcepts through their common concept. Therefore, it is possible to compare connected subconcepts groups, as illustrated by Figure 3.5.

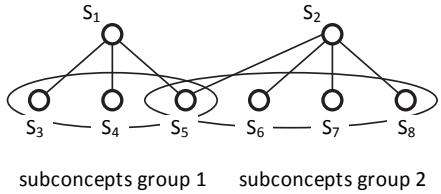


Figure 3.5. *A closed subset*

By checking the accuracy of the substructure contained in a *closed subset*, it can be enquired that if the common subconcept (S_5) of group 1 and group 2 is connected to superconcepts S_1 and S_2, why the other members of group 1 (S_3 and S_4) are not related to superconcept S_2. In addition, why are the members of group 2 (S_6, S_7, and S_8) not related to S_1?

3.2.3 Large closed subset (LCS)

The smallest size of a *closed subset* is 1 (the number of upper-level *synsets*) x 1 (the number of lower-level *synsets*). The size of *closed subsets* may vary as the biggest size in

different wordnets. In many cases, LCS seems to be he particular feature of the hierarchical structure that links different hierarchical structures started from unique beginners. The largest *closed subset* we have found in plWordNet (Polish Wordnet) (Maziarz et al., 2012) is 30,794 x 4,463. It consisted, among other things, of 142 *root synsets* (Lohk et al., 2014c). This number is undoubtedly too big. (It may be remembered that at the beginning of Princeton WordNet creation, only 25 topmost concepts were used for noun hierarchy and 14 topmost concepts for verb hierarchies.)

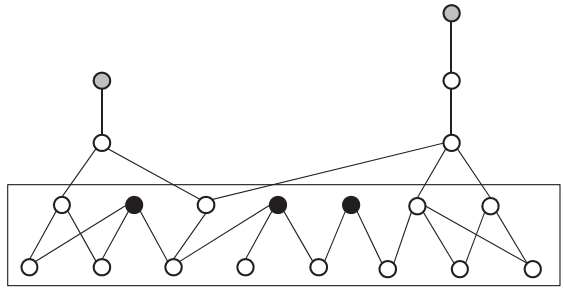


Figure 3.6. *An artificially constructed "large" closed subset*

In Figure 3.6, there is an artificially created "large" *closed subset* highlighted by a rectangle. Nodes filled with black colour denote unique beginners (*root synsets*) in a closed set. Grey nodes indicated unique beginners outside of *closed subsets* but related to one. The *large closed subset* as a particular feature of the wordnet hierarchy seems to indicate the accuracy of the wordnet hierarchical structure. Thus, its chief benefit is detecting the general state of a wordnet structure as regards its accuracy (Lohk et al., 2014c).

3.2.4 Root synset in a closed subset

It is not easy to use instances of *large closed subset* for detecting concrete errors in a wordnet hierarchical structure but it is possible, as it is proposed in (Lohk et al., 2014b). It is far simpler to discover small *closed subsets* that consist of unique beginners (*root synsets*) in the upper level of bipartite graphs (*closed subsets*). The idea of that approach is to see which *root synsets* are in same level with non-*root synsets*. Most likely, there are errors which express itself as unfinished work because a *root synset* and a non-*root synset* cannot be on the same concept level. The solution to that is either to add a higher-level concept to a *root synset* or to connect it to pre-existing higher-level concepts. Figure 3.7 illustrates a *closed subset* with a *root synset*.

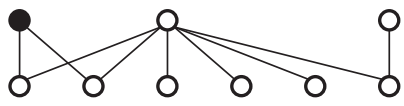


Figure 3.7. *Root node (filled node) in a closed subset*

3.2.5 Dense component

The *dense component* pattern provides the opportunity to uncover substructures where, due to the *multiple inheritance*, the density of the interrelated concepts in the semantic hierarchy is higher (Lohk et al., 2014a), (Lohk et al., 2014b). This substructure (subgraph) consists of two *synsets* (nodes) with at least two identical parents (it corresponds to *complete bipartite graph*). The overall size of an instance of a *dense component* depends on how many *synsets* (nodes) with at least two parents are interconnected through the *multiple inheritance* and/or same parents. However, let us explain its detection algorithm in a more concrete way. If we assume that there exists a set of nodes with many parents. Some of them form the *dense component*s and some do not. One set that forms a *dense component* is in Figure 3.8.

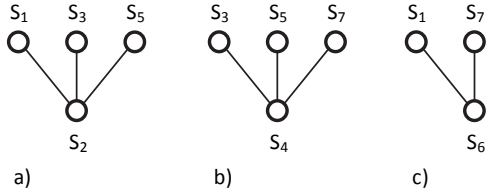


***Figure 3.8**. Subgraphs, synsets with many parents*

If we compare subgraph (a) to subgraph (b), then it emerges that they have two identical parents – S_3 and S_5. At the same time, subgraph pairs (a) - (c) and (b) - (c) do not meet the criteria of *at least two identical parents*. Thus, (a) and (b) with nodes from S_2 to S_5 and their relations form the *complete bipartite graph*. However, after joining subgraphs (a) and (b), subgraph (c) also fits into this component while S_6 has two parents identical with those of (a) and (b). The result of this *dense component* is depicted in Figure 3.9.

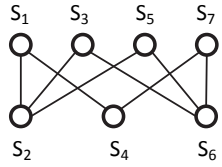


***Figure 3.9**. Dense component*

Figure 3.9 contains an example of the *multiple inheritance* caused by *regular polysemy*. This is due to the fact that the polysemic S_2 and polysemic S_4 have a simultaneous connection to S_3 and S_5. *Synset* S_4 does not fit into the *regular polysemy* category because there is no other *synset* that has at least two identical parents with *synset* S_4.

In the evaluation process, the expert linguists/lexicographer has to check whether the *multiple inheritance* is justified. In our experiment (Lohk et al, 2014b), we found that among the other errors, the *multiple inheritance* was not justified in most cases.

(Based on Figure 3.9, the expert could also ask that if S_2 and S_6 are part of the same *regular polysemy*, why S_4 has no connection to S_3 and S_5. On the other hand, it can be queried why S_4 has a connection to S_1 and/or S_7 if it is not part of the *regular polysemy*.)

3.2.6 Heart-shaped substructure

According to Figure 3.10, in a *heart-shaped substructure* pattern, two nodes (S_2 and S_4) have a direct connection through an identical parent (S_3) and an indirect connection through a semantic relation (S_5 – S_1) that links their second parent.

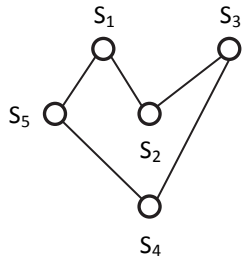


Figure 3.10. *Heart-shaped substructure*

In spite a *synset* having more than two parents, only two of them can simultaneously be part of the *heart-shaped substructure*. Thus, for a *synset* with three or more parents, all the combinations of the *synset* with two parents are detected and every combination may be a part of a *heart-shaped substructure*. Figure 3.11 (a) contains an example of a *synset* with three parents, part (b) shows all the combinations of a *synset* with two parents of part (a). The number of combinations can be calculated using formula $\frac{n\,(n-1)}{2}$, where n is the number of parents of a *synset*. For example, for five parents, *synset* n equals 10.

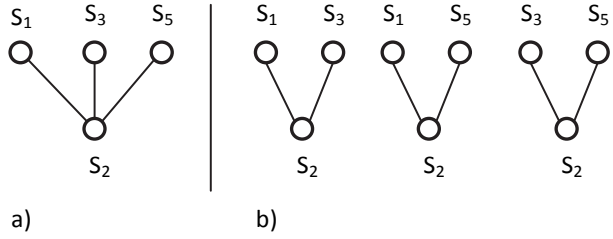


Figure 3.11. *Combinations of a synset and its parents*

Linguists from Princeton University used the *heart-shaped substructure* for checking Princeton WordNet (version 3.1). They noticed that this pattern is helpful for detecting wrong semantic relations, mostly *role* and *type* relations (Lohk and Võhandu, 2014), (Lohk et al., 2014b). Naturally, it is not right to assume that different language wordnets will contain the same types of errors because they have

different building bases and also culture and region specific concepts. Nevertheless, it is a well-known fact that Princeton WordNet does not contain *type* and *role* semantic relations (Atserias et al., 2005), (Martin, 2003).

The latest investigations of the author of this thesis show that wordnets have a pattern that could be termed a *complete heart-shaped substructure* (Figure 3.12). The difference lies in the fact that both *synsets* (S_2 and S_4) have an indirect connection with semantic relations (S_5 – S_1 and S_7 – S_3) which link both their parents. However, this substructure has not been used to check and validate the wordnet hierarchical structure but it is assumed that it would detect at least the same kinds of errors as a *heart-shaped substructure*.

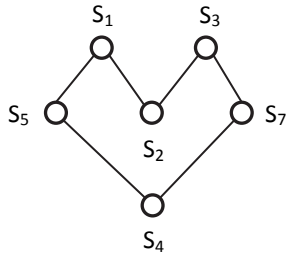


Figure 3.12. *A complete heart-shaped substructure*

3.2.7 Substructure that considers the content of synsets

This substructure differs from others because it takes into account the content of the *synsets*. More precisely (Figure 3.13), for every pattern of that kind, we are interested in the *hypernym* (S_3) that the member (*lexical unit*) includes in at least two of the compound words of its *hyponyms* or in part of multiword(s) (S_2, S_4, S_6, S_8, S_{10}). Examples of these two cases are:

1) Hypernym *paper* (S_3) and its hyponym *newspaper*

2) Hypernym *paper* and its hyponym *roofing paper*

Secondly, at least one *hyponym* (S_2) has (an) additional parent(s) (S_1). Using the abovementioned example – *roofing paper* (S_2) has the hypernym *roofing material* (S_1).

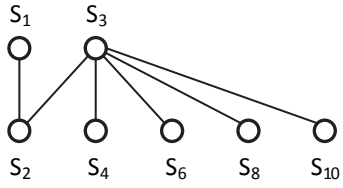


Figure 3.13*. Substructure that considers the content of synsets*

The Estonian language is similar to the German language, in that there are numerous compound words. Therefore, in the Estonian Wordnet, the member of the *hypernym*

synset (S_3) mainly connects to the member of the *hyponyms* that is a compound word (e.g. newspaper). On the other hand, in the English wordnet, the member of the *hypernym synset* chiefly connects to the member of the *hyponyms* that is a multiword (e.g. roofing paper).

To validate this pattern, the expert linguists/lexicographers must consider has why S_1, whilst connected to S_2, has no connections to S_4, S_6, S_8 or S_{10} while they have connections to S_3 as well as S_2. This enquiry helps to make a decision regarding the inconsistencies that this pattern may have. As regards the Estonian Wordnet, we found that sometimes this pattern points to a situation where a superordinate had inappropriate meaning (S_1). For example, *boa* as a snake (S_2) had a hypernym *scarf* (S_1) (see Figure 4.9 in Chapter 4).

3.2.8 Connected root synsets

This pattern is different from others and required additional information about vertices and edges. In Figure 3.14, an edge connects two vertices, if two hierarchies that started from these vertices have at least one common item. In addition, this pattern represents a) *a top view* b) *in global perspective* for the particular part-of-speech hierarchies. The size of this pattern can vary largely.

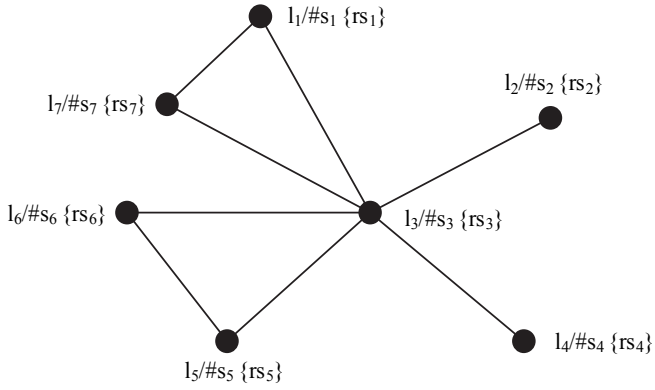


Figure 3.14. *Connected root synsets*

The following sets are defined: #L={#l_1, #l_2, #l_3, #l_4, #l_5, ... #l_n}, #S = {#s_1, #s_2, #s_3, #s_4, #s_5, ..., #s_n}, RS = {rs_1, rs_2, rs_3, rs_4, rs_5, ..., rs_n}, where L is the number of the maximum levels for certain *root synset* hierarchies, #S is the number of *synsets* in a certain hierarchy, and RS is the set of *root synsets* for a certain part of speech. Every item in RS can be depicted as vector:

$rs_i = \langle l_i, \#s_i \rangle$

Where

l_i is the number that indicates the depth of the rs_1 hierarchy

$\#s_i$ is the total number of subordinates in the rs_i hierarchy

Every edge is described as a vector:

$e(rs_i, rs_j) = <\#cs_{i,j}, level_i, level_j, first\text{-}cs_{i,j}>$

Where

$\#cs_{i,j}$ is the number of common *synsets* of rs_i and rs_j hierarchies

l_i is a number that shows on which level the common *synset* is in rs_i (*root synset* i)

l_j is a number that shows on which level the common *synset* is in rs_j (*root synset* i)

$first\text{-}cs_{i,j}$ is the name of the first common *synset* of the two hierarchies (rs_i and rs_j).

In our experience, sometimes the number of *root synsets* may exceed 500. Even if it is up to 100, this approach is not applicable. In that case, instead of a text label, it is reasonable to use numbers and represent all the textual information in a table.

The benefits of this pattern is that it shows for a particular part-of-speech hierarchy:

1) How many hierarchies there are
2) Which hierarchies thread to others
3) Which hierarchies are separated
4) How big or small the hierarchies are
5) Whether there are *root synsets* that are too specific
6) On which level the common *synsets* appear.

3.3 AN OVERVIEW OF THE TYPICAL ERRORS CONNECTED TO TEST PATTERNS

The aim of this section is to give a compact overview of the proposed test patterns and typical errors they help to detect. In addition to our test patterns, *short cut* and *ring* test patterns are included here, mentioned by (Fischer, 1997), (Liu et al., 2004), (Richens, 2008).

There is no doubt that each test pattern may lead to a different error, even to ones not mentioned when describing each test pattern. Some examples are *a redundant*[40] *or missing sense*, *redundant or missing semantic relation*, *redundant or missing lexical unit in* a *synset* and others. It is important to mention that the *typical error(s)* are not clear for every test pattern. This is mainly because every wordnet has different building and extending bases (Section 1.3) and also culture and region specific concepts. Secondly, substantial feedback only exists for some test patterns. Despite that, in our experience, all the test patterns uncover different errors (Section 3.3). In

[40] Sense may be redundant due to a *fine-grained* problem

the *comment* part, the content of *typical error(s)* is put forth by substantiating our belief in why this error is typical for that kind of a test pattern.

- **Short cut** may appear to be an instance of *multiple inheritance* but yet it is not. Instead, it refers to the case where one *synset* has a connection to another one both directly and indirectly.
  Typical error: redundant link.
  Comment: this pattern always refers to the error of redundant link (Fischer, 1997), (Liu et al., 2004), (Richens, 2008).
- **Ring** arises when a subordinate with many parents has a superordinate via two branches.
  Typical error: in some cases, a *synset* with many parents cannot inherit opposite properties from different *hypernyms*.
  Comment: (Liu et al., 2004) regard rings abnormal, in particular when both parents in the ring originate from same the domain category.
- **Synsets with many roots** – a substructure where one *synset* has a connection to two different *root synsets*.
  Typical error: many root synsets for one synset are not justified, some root synsets are too specific to be root synsets.
  Comment: Similar to the pattern of *connect root synsets*. Therefore, similar errors are expected to be seen. Instead of a top view, the side view is used. In addition, only two *root synsets* are part of that pattern.
- **Closed subset** is a coherent bipartite graph in two sequential levels of the wordnet hierarchical structure.
  Typical errors: *synset* has a connection to a specific and a general concept, wrong or missing senses, wrong or missing relations.
  Comment: Karin Kungla from University of Tartu tested closed sets in her BA thesis on Estonian Wordnet (version 60); in two papers, Kadri Vare from University of Tartu analysed about 20 *closed subset* instances (Lohk et al., 2012a), (Lohk et al., 2012b).
- **Large closed subset** is a special case of the *closed subset*, representing *the largest subset*.
  Typical errors: indicates the general accuracy of a wordnet hierarchical structure.
  Comment: An experiment for the paper of (Lohk et al., 2013). We proposed a fast algorithm to minimize their crossing number in a bipartite graph. We found *the largest closed subset* for seven different wordnets. Three of them can be classified as very large ones. The second experiment is described in (Lohk et al., 2014c).

- **Root synset in the closed subset** is a *closed subset*, which contains a *root synset* at its upper level.
  Typical errors: unfinished work; a *root synset* is too specific to be a *root synset*; it belongs to another hierarchy.
  Comment: We made some examples for presentations at various science conferences (6th Global WordNet Conference in Matsue, Japan, January 9-12, 2012; EACL 2012 Joint Workshop of LINGVIS & UNCLH, Avignon, France, April 23-24, 2012; Information and Software Technologies: 20th International Conference, ICIST 2014, Druskininkai, Lithuania, October 9-10, 2014) and finally published it in (Lohk et al., 2014b).
- **Dense component** is a substructure that contains at least two *synsets* with two identical parents (for complete definition see Section 3.2.5).
  Typical error: the *multiple inheritance* is not justified or has to be expanded
  Comment: We conducted an experiment on Estonian Wordnet (version 66) and the aforementioned *typical error* was indeed the most frequent case (Lohk et al., 2014a).
- **The heart-shaped substructure** is a substructure where two *synsets* have a direct connection through a common parent and an indirect connection with a second parent through a semantic relation.
  Typical error: points to the wrong semantic relationship
  Comment: Linguists from Princeton University used a *heart-shaped substructure* for checking Princeton WordNet (version 3.1). They noticed that this pattern is helpful for detecting the wrong semantic relations, mostly *role* and *type* relations (Lohk and Võhandu, 2014), (Lohk et al., 2014b).
- **Substructure that considers the content of synsets** – here, the *hyponymy* relation and *multiple inheritance* is included when the *lexical unit* of a *hypernym* is part of a compound word or a multi-word of a *lexical unit* of a *hyponym*.
  Typical error: the wrong semantic relationship
  Comment: The first quick overview of Estonian Wordnet (version 68) surprisingly uncovered many *synsets*, which did not fit into that particular place in structure – so-called careless mistakes (Lohk et al., 2014b).
- **Connected root synsets** represent a top view in the global perspective.
  Typical errors: hierarchies that are too small; concepts that are too specific for the *root synset*; too many *root synsets* and too many connections between them.
  Comment: An experiment for a poster presentation at the Annual Applied Linguistics Conference (April 19-19, 2013, Tallinn). The experiment involved Princeton WordNet (version 3.1) and Estonian WordNet (version 65).

3.4 CONCLUSIONS

There is nothing new in applying certain substructures of specific nature (test patterns) for checking and validating the semantic hierarchies of a wordnet. However, so far only a few authors have used them (Fischer, 1997), (Liu et al., 2004) and (Richens, 2008) or have suggested their use (Koeva et al., 2004), (Šmrz, 2004), (Čapek, 2012). Their infrequent use may arise from the situation that most of these inconsistencies are avoidable with a wordnet management system, such as *cycles, null graphs, loops, short cuts*.

In this chapter we described substructures, which are yet undiscovered in the semantic hierarchies of wordnet and which contain at least one *multiple inheritance*. However, while other authors have also referred to substructures with *multiple inheritance,* two patterns (*short cut* and *ring*) are inspired by the authors (Fischer, 1997), (Liu et al., 2004) and (Richens, 2008). Every test pattern is associated with typical errors they may help to discover.

Even though every test pattern in this chapter is associated with typical errors they may help to discover from the semantic hierarchies of wordnet, thorough experiments were performed for only two test patterns – the *heart-shaped substructure* and the *dense component*. Both of these patterns yielded very good results. At first, linguists from Princeton University applied the *heart-shaped substructure* to Princeton WordNet (version 3.1). They found that in most cases this pattern refers to the wrong semantic relation, where instead of a *hypernymy* relation, the *role* or *type* relation should have been used (Lohk and Võhandu, 2014). Secondly, linguist Heili Orav from the University of Tartu employed the *dense component* in Estonian Wordnet (version 66) (case study in Section 4.2). As a result, in most of the discovered cases the *regularity of multiple inheritance* (Section 1.2.8) was not justified (Lohk et al., 2014a).

In conclusion, since every wordnet has a different building and extending basis, we cannot strongly claim that every wordnet will only yield the errors described here alongside every test pattern. In our experience, by using different test patterns, the expert lexicographer/linguist may discover a wide range of different types of errors. In very rough terms, all the error corrections can be traced back to these acts:

- merging many *synsets* or dividing one
- deleting a *synset*
- adding or removing a *lexical unit* of a *synset*
- adding or removing a semantic relation

Subsequently, our test patterns are put to action.

4. PATTERNS IN ACTION

"When building large-scale lexical/semantic resources, subsequent – or better, simultaneous – validation of content is essential" – Dietrich H. Fischer

The test patterns used in the validation of the wordnet semantic hierarchy primarily indicate the possible errors it may contain. Despite the fact that it is not difficult for non-experts to detect most of the errors, in everyday practice, the lexicographer validates all the instances of test patterns and corrects them if need be.

Until now, test patterns have been applied to EstWN after the release of each new version. It means that the correction and expansion of the latest EstWN version will be conducted almost simultaneously. Nevertheless, for the most recent version of EstWN, test patterns were used in the validation before it was made publicly available online[41].

This chapter describes the instance errors for each test pattern, primarily utilizing EstWN examples, but there are also two examples from PrWN. All the cases are only related to verb and noun hierarchies and *hypernymy* relations.

Finally, a case study of the *dense component* is presented, which is based on a paper of ours (Lohk et al., 2014a). In this context, we look at what operations a lexicographer performs to correct the 121 *dense components* of EstWN Version 66 as well as how these corrections reduced the number of *dense component* instances and *multiple inheritance* cases in the following version 67.

This chapter is mainly based on the papers "New Test Patterns to Check the Hierarchical Structure of Wordnets" (Lohk et al., 2014b) and "Dense Components in the Structure of Wordnet" (Lohk et al., 2014a). It is also partly based on the following papers: "How to create order in *large closed subsets* of wordnet-type dictionaries" (Lohk et al., 2013), "Independent Interactive Testing of Interactive Relational Systems" (Lohk and Võhandu, 2014) and "Some structural tests for Wordnet with results" (Lohk et al., 2014c)

4.1 EXAMPLES OF TEST PATTERNS

This section studies examples of test pattern usage. The main examples used are from different EstWN versions, but two examples originate from Princeton WordNet Version 3.1. In this context, the content of every test pattern is explained again shortly and the errors that every instance contains are described.

In everyday practice, a lexicographer validates the instances of test patterns and makes corrections, if required. As regards these examples, the author of this thesis

[41] http://www.cl.ut.ee/ressursid/teksaurus/index.php?lang=en

provides an assessment of every example, using among other things online wordnets – EstWN[42] and Princeton WordNet[43].

While two test patterns have been inspired by other authors (Fischer, 1997), (Liu et al., 2004) and (Richens, 2008), these are placed at the start.

In the EstWN examples, every *synset* is equipped with the equivalent *synonyms* from Princeton WordNet Version 1.5[44] and begins with an abbreviation "(Eq_s)". If the equivalent *synonyms* are unknown, free translation has been used. For instance, in Figure 4.2, the first three *synsets* have equivalent *synonyms* but the bottommost entails a free translation.

To save space, examples are presented in a slightly different manner from the output of programs. Notably, they do not contain glosses (*synset* definitions). Some of the examples of test pattern instances delivered by our programs (see Chapter 5) are available on a special webpage[45].

4.1.1 Short Cut

The *short cut* pattern represents the situation where a lexical concept (*synset*) has two parents but at the same time, it is not ambiguous. In Figure 4.1, the *synset* {club soda_1, mineral water_1 ...} is one such example. That is to say, the *synset* {club soda_1, mineral water_1} has a relationship with {beverage_1, drink_2, potable_1}. However, the latter is merely the more general concept {mineraalvesi_4} with the equivalent *synset* {club soda_1, mineral water_1 ...}, which is unnecessary information for {club soda_1 ...} and therefore, the dotted line is redundant.

Secondly, it does not directly concern the EstWN hierarchy, but nevertheless {mineraallvesi_4} and {soodavesi, mineraalvesi, selter} have the same equivalent *synset* {club soda_1, mineral water_1 ...}. This may refer to the case that both of these *synsets* are incorrectly mapped or there are no two different concepts for {mineraalvesi_4 ...} and {sooda vesi_1 ...}. However, in the present case, the PrWN (Version 3.1) has two different concepts – {mineral water_1} and {soda water_1, carbonated water_1, club soda_1, seltzer_2, sparkling water_1}.

[42] http://www.cl.ut.ee/ressursid/teksaurus/teksaurus.cgi.en
[43] http://wordnetweb.princeton.edu/perl/webwn
[44] EstWN is currently mapped to Princeton WordNet Version 1.5 (11.05.2015)
[45] https://sites.google.com/site/instances2015/

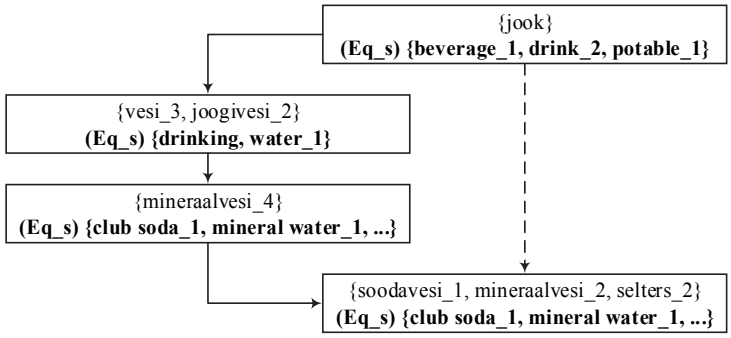


Figure 4.1. *A short cut, EstWN (version 70)*

4.1.2 Ring

This pattern category contains both *symmetric rings* (Figure 4.2) as well as *asymmetric rings* (Figure 4.3). In the checking procedure, the first question for the lexicographer is "can it be true that lexical concepts with two parents simultaneously belong to both classes?" According to Figure 4.2, the answer is undoubtedly "yes". That is to say, "soup bowl" is simultaneously {bowl} and "dishware". The answer for Figure 4.3 is the inverse "no", since it is impossible for "stone marten" to simultaneously be "weasel" as well as {bird}. For corrections, according to the latest version of EstWN (version 71), the relation {bird} is removed and the concept "marten" is added between "stone marten" and "weasel".

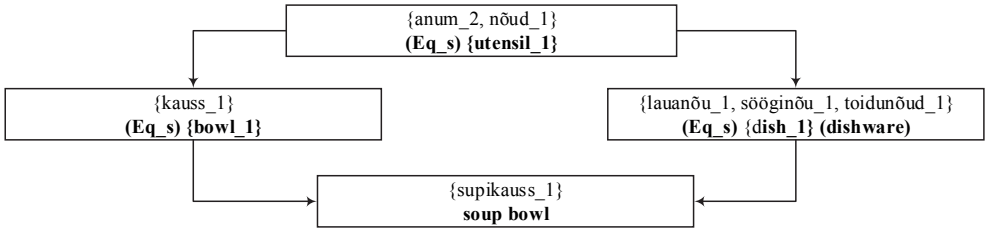


Figure 4.2. *A symmetric ring, EstWN (version 68)*

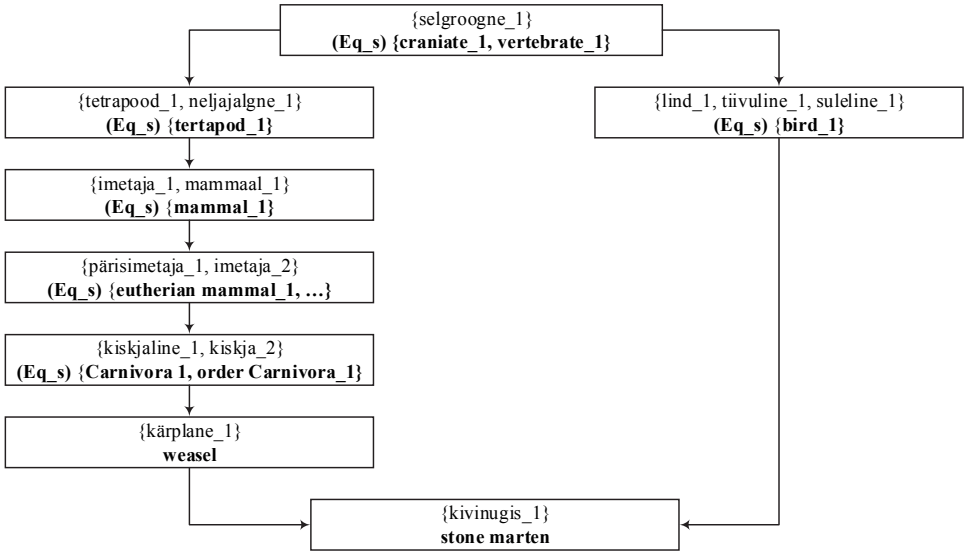


Figure 4.3. *An asymmetric ring, EstWN (version 69)*

The symmetric ring pattern appears to be especially beneficial when both branches only have one concept as in Figure 4.2 – {bowl_1} and {dish_1}. In that case, the lexicographer has to compare if they are concepts from different levels. As depicted in Figure 4.2, the concept {dish} seems to be superordinate to {bowl}. However, as the correction, the lexicographer has removed the relationship between {dish_1} and "soup bowl". Nevertheless, {bowl_1} and {dish_1} are on the same concept level in the latest EstWN version. Why?

4.1.3 Synset with many roots

Quite a similar pattern to the previous rings is the *synset* with many roots. This pattern differs from the former one by its unconnected branches. On the one hand, it means that some of the detectable errors are similar to rings and on the other hand, it is capable of discovering errors related to *root synsets*.

Figure 4.4 demonstrates how one *root synset* is *a dangling uplink*[46] – "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 too lower-level concept.

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

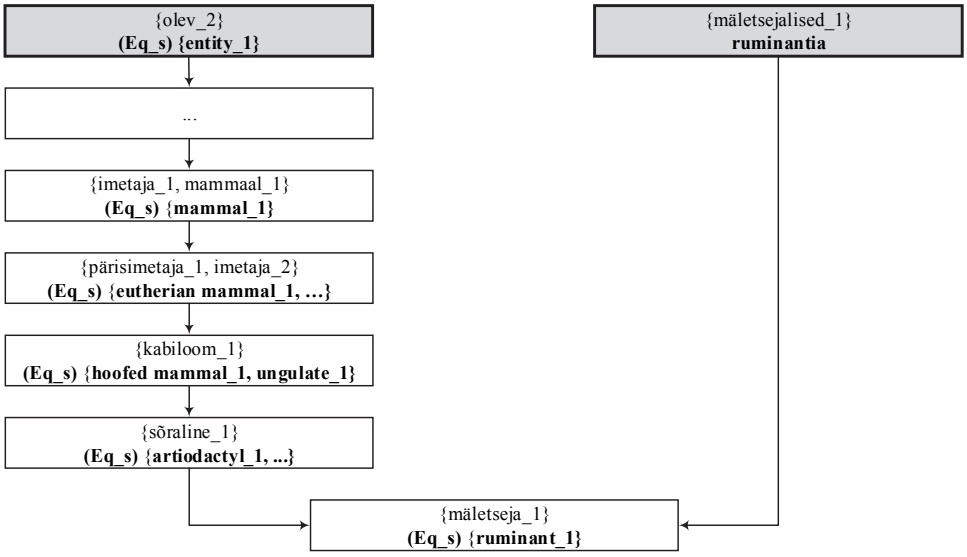


Figure 4.4. *Synset with many roots, EstWN (version 71)*

The *root synset* "ruminantia" is a taxon, i.e. it presents 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 of "ruminantia".

4.1.4 Root in the closed subset

The ordinary checking procedure for the instance of *closed subset* test pattern begins with separating the subgroups of subordinates, as in Figure 4.5 – I group, II group, and III group. The lexicographer should aim to distinguish them by sense and ask why common concepts of two groups connect them, i.e. whether this link is justified.

When a *closed subset* contains a *root synset*, the usual solution is be either:

- Connecting that *root synset* to a higher level concept from outside the *closed subset*, or
- Connecting that *root synset* to a higher level concept from inside the *closed subset*.

However, as shown in Figure 4.5, generally these two possible actions may not be sufficient for all the corrections of that substructure.

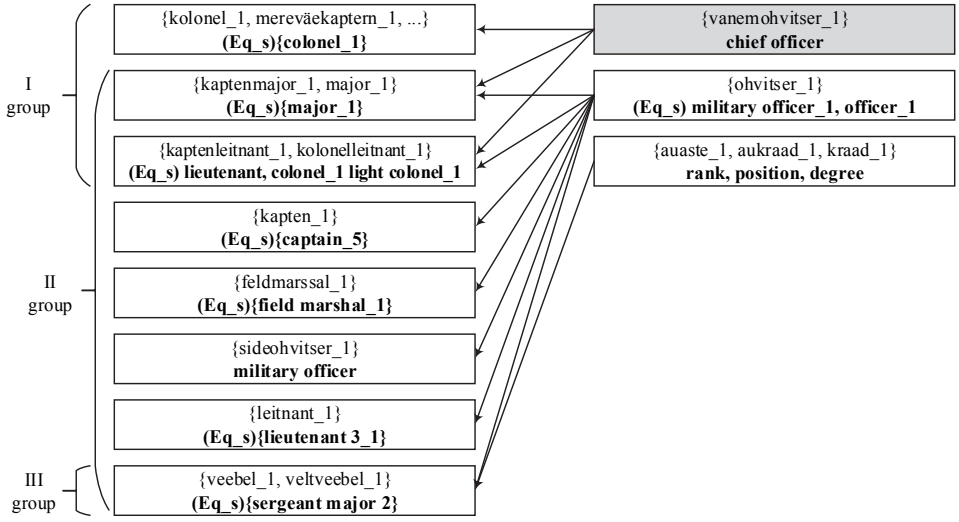


Figure 4.5. *Root synset in a closed subset, EstWN (version 62) (rotated 90 degrees)*

In the latest version of EstWN, "chief officer" is located under {military officer_1, officer_1}. Nevertheless, there are several other problems.

- Firstly, all 8 subordinates belong to the superordinate "rank, position, degree".
- Secondly, on the same level as "chief officer" there should be "subaltern, petty officer" and "brass hat" as there are three types of military officers – "chief officer", "subaltern, petty officer" and "brass hat".
- Thirdly, {military officer_1, officer_1} is a "rank, position, degree".
- Fourthly, it emerged that even though "brass hat" is included in EstWN, its *synset* also contains its subordinate – {kõrgem ohvitser, kindral} – "brass hat, general".
- Fifthly, at least one subordinate simultaneously contains the rank of Navy forces and land forces ({kaptenmajor_1, major_1}) and at least one does not ({kapten_1}).
- Sixthly, not all the ranks are included in EstWN. For instance, "vanemleitnant" (chief lieutenant),
- Seventhly, veebel_1 and veltveebel_1 ({sergeant major_2}) are different-level ranks from the class of non-commissioned officers.

4.1.5 Large closed subset

The *large closed subset* is possible due to the *multiple inheritance* cases in the particular concepts level in the wordnet hierarchy. In Figure 4.6, there is a fragment of the largest *closed subset* of PrWN (version 3.1). It has 1,064 *hyponyms* (row labels) and 126 *hypernyms* (column labels). According to our example, there are about 16% (170) of *hyponyms* with more than one parents.

	{food (%1:03:00::),...}	{b (%1:27:01::), b complex, b vitamin,...}	{peroxide (%1:27:01::)}	{blanching agent (%1:27:00::), bleach,...}	{body covering (%1:08:00::)}	{hypochlorite (%1:27:00::)}	{frame (%1:08:00::), skeletal system,...}	{greenhouse emission (%1:27:00::),...}	{antimicrobial (%1:06:00::),...}	{cationic detergent (%1:27:00::),...}	{fluorocarbon (%1:27:00::)}	{fluoride (%1:27:00::)}	{connective tissue (%1:08:00::)}	{solution (%1:27:00::)}	{atomic number 6 (%1:27:00::), c,...}
{culture medium (%1:27:00::),...}	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
{choline (%1:27:00::)}	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
{inositol (%1:27:00::)}	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
{aneurin (%1:27:00::),...}	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
{antipernicious anemia factor (%1:27:00::), coba	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
{hepatoflavin (%1:27:00::), lactoflavin,...}	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
{adermin (%1:27:00::), pyridoxal,...}	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
{folacin (%1:27:00::), folate,...}	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
{niacin (%1:27:00::),...}	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
{biotin (%1:27:00::),...}	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
{bleaching powder (%1:27:00::),...}	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
{benzoyl peroxide (%1:27:00::)}	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
{hide (%1:05:00::), pelt,...}	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
{protective covering (%1:05:00::)}	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
{exuviae (%1:08:00::)}	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
{hair (%1:08:00::)}	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
{headful (%1:08:00::)}	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
{epicranium (%1:08:00::)}	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
{calcium hypochlorite (%1:27:00::)}	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
{exoskeleton (%1:08:00::)}	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
{sodium hypochlorite (%1:27:00::)}	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0
{sulfur hexafluoride (%1:27:00::),...}	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
{endoskeleton (%1:08:00::)}	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
{hfc (%1:27:00::),...}	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
{perfluorocarbon (%1:27:00::),...}	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
{cutis (%1:08:00::), skin,...}	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
{chlorine water (%1:27:00::)}	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0
{cetrimide (%1:27:00::)}	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
{boron trifluoride (%1:27:00::)}	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
{hydrogen fluoride (%1:27:00::)}	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
{stannous fluoride (%1:27:00::)}	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
{tetrafluoroethylene (%1:27:00::)}	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
{cfc (%1:27:00::),...}	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
{areolar tissue (%1:08:00::)}	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
{bone (%1:08:00::),...}	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Figure 4.6. *A fragment of a large closed subset (1064x126), PrWN (version 3.1)*

Beneficially, the *large closed subset* test pattern indicates the accuracy (or inaccuracy) of the whole hierarchical structure of wordnet. Too many contrast concepts among the *hypernyms* or *hyponyms* may however arise suspicions. For example, in Figure 4.6, there are the contrastive *hypernyms* such as {food, ...}, {body covering}, {greenhouse emission, ...} and {atomic number 6}.

The *large closed subset* does not specifically indicate a certain (small) place where a possible error may appear. Instead, it allows the lexicographer to follow the line of "1" and see how this "chunk" is formed.

A good way of studying these instances is to save them in a spreadsheet application and freeze the column and row fields and then to scroll through following the "1" line. The example of Figure 4.6 is saved in Google Spreadsheet[47].

4.1.6 Dense component

The *dense component* pattern represents the substructure of a wordnet hierarchical structure with a high concentration of interconnected *synsets*. This pattern contains at least two ambiguous concepts (as in Figure 4.7 {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* (Section Figure 1.11).

The lexicographer has to check:

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

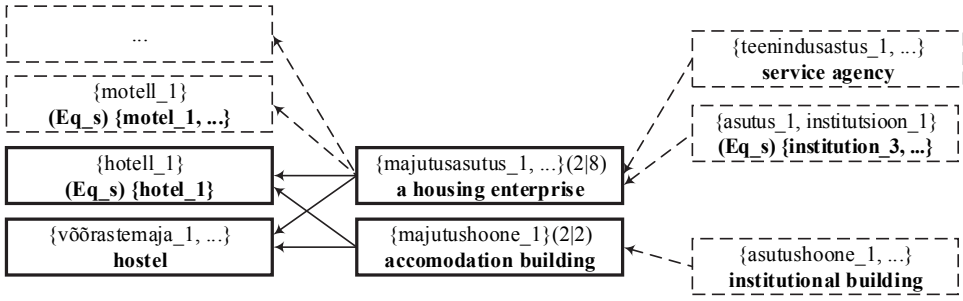


Figure 4.7. *A dense component, EstWN (version 66) (rotated 90 degrees)*

In order to better understand the semantic field of the dense component in Figure 4.7, the *synsets* with dotted lines are additional information to the *dense component* (*synsets* with bold lines) for more clearly grasping its content. The first number after the *synset* in brackets indicates the number of subordinates inside the *dense component*.

[47] https://sites.google.com/site/instances2015/large-closed-subset

The second number in brackets displays the number 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 4.7, {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".

In the latest version of EstWN, it emerged that {hotel_1} and "hostel" are no longer connected to building through a *hypernymy* relation. (Instead, it has a connection through *near_synonymy*.) Meanwhile, in PrWN, {hotel_1} is only a building and {hostel_1} is its subordinate.

For a solution, let us look at another concept similar to a motel, hotel, and hostel – the hospital. EstWN organizes this concept into two *synsets*. The first one is in the meaning of a *medical institution* and the second one in the sense of a *medical building*. A similar idea is followed in PrWN. Thus, in both wordnets, the *hospital* is related to an *institution* as well as a *building*. According to this example, it is advisable to organize the *hotel*, *motel* and *hostel* in a similar manner.

4.1.7 Heart-shaped substructure

The *heart-shaped substructure* pattern describes the substructure in the wordnet hierarchy where two *synsets* (in Figure 4.8, {homoepathy_1} and "mud cure, mud treatment") along with their two parents are interconnected due to a common parent ({curative_1, cure_1}) and through a *hypernymy* relationship between another one of their parents ({naturopathy_1} and {alternative medicine_1, ...}).

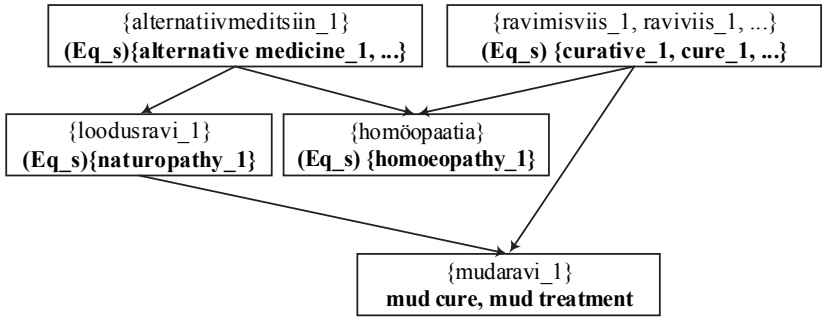


Figure 4.8. *Heart-shaped substructure, EstWN (version 67)*

In the report file on the instances of a *heart-shaped substructure*, we deliver 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 4.8 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*. Despite that there is no such 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 PrWN pointed to the situations where instead of a *hypernymy* relation there should have been a *role* or *type* relation.

4.1.8 Substructure that considers the content of synsets

(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 *synsets* 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 4.9, 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 connection to the other subordinates. Upon checking this additional concept ({scarf_1}), it emerges that this is totally unsuitable because while the {boa_1} is a *serpent*, the *scarf* is a *garment*. However, the *scarf* is still related to the *boa*, but in a different meaning {boa_2, feather boa_1}.

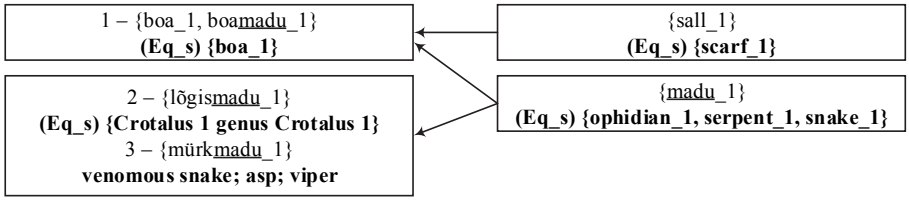


Figure 4.9. *Substructure that considers the content of synsets, EstWN (version 69)*

Looking at Figure 4.10, we see two "extra" concepts – {kindergarten_1} and {teaching method_1}. When {kindergarten_1} is connected to "outdoor games" wrongly, then the connection of {teaching method_1} is justified.

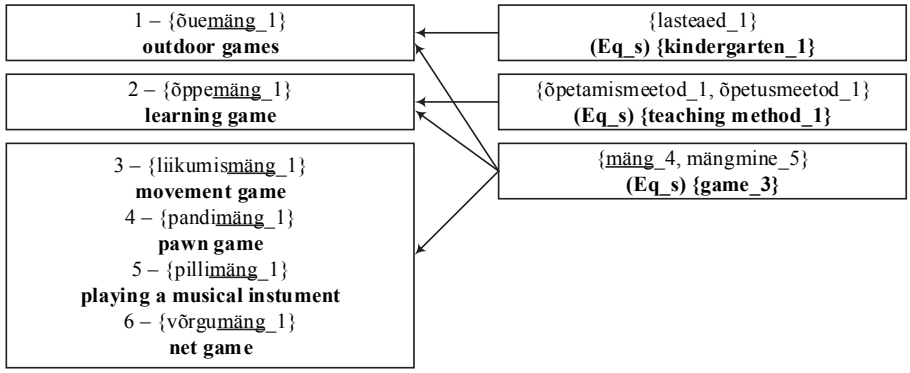


Figure 4.10. *Substructure regarding the content of synsets, EstWN (version 69)*

In addition, there are two other changes in this substructure in EstWN (version 70):

- "Learning game" no longer has a relation to {game_3}, and
- "Playing a musical instrument" is removed from EstWN

4.1.9 Connected roots

The test pattern of *connected roots* covers different hierarchies through *multiple inheritance* cases. Every node as a *unique beginner* is equipped with the number of hierarchy levels and the number of subordinates in the same hierarchy. The first number of the edge label indicates the number of common subordinates for two hierarchies. The next two numbers separated by " | " denote the hierarchy levels where the first common concept is located in both hierarchies.

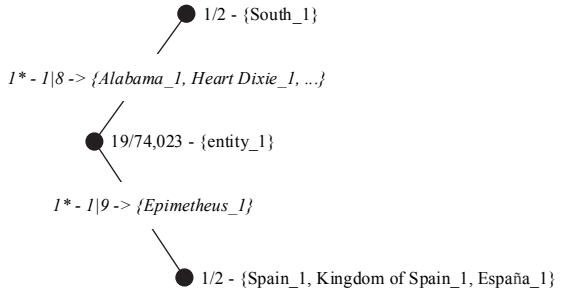


Figure 4.11. *Connected noun roots, PrWN (version 3.1)*

In Figure 4.11, there is only one large hierarchy with the unique beginner {entity}. It has a 19-level hierarchy and 74,023 subordinates. On the contrary, the two other hierarchies ({South_1} and {Spain_1 ...}) are minuscule. They are both 2-level

hierarchies. The edge labels reveal that the common concepts of both hierarchies are on the first levels in both of the smaller hierarchy cases. Possible problems occur due to:

- Hierarchies being too small ({South_1} and {Spain})
- The common concept is on the first or second level of the hierarchy ({Alabama_1, ...} and {Epimetheus_1})
- Unique beginners are concepts from different concept levels {entity_1} and {South_1}
- It is clear that common concepts cannot belong to both hierarchies (in Figure 4.12 {freeze_7} as feel chilly, which cannot have the meaning of {do_6, execute_3, perform_1})

The substructure in Figure 4.11 has been changed in current PrWN version. {Alabama_1 ...} has still two parents, but instead of {entity_1}, it is connected to {American States_1}, which in turn is related to the unique beginner {state_1, province_1}. Moreover, {Epimetheus_1} is now connected to only one parent – {Titan_2}, which is also a unique beginner. Both unique beginners are too specific to be the highest concepts.

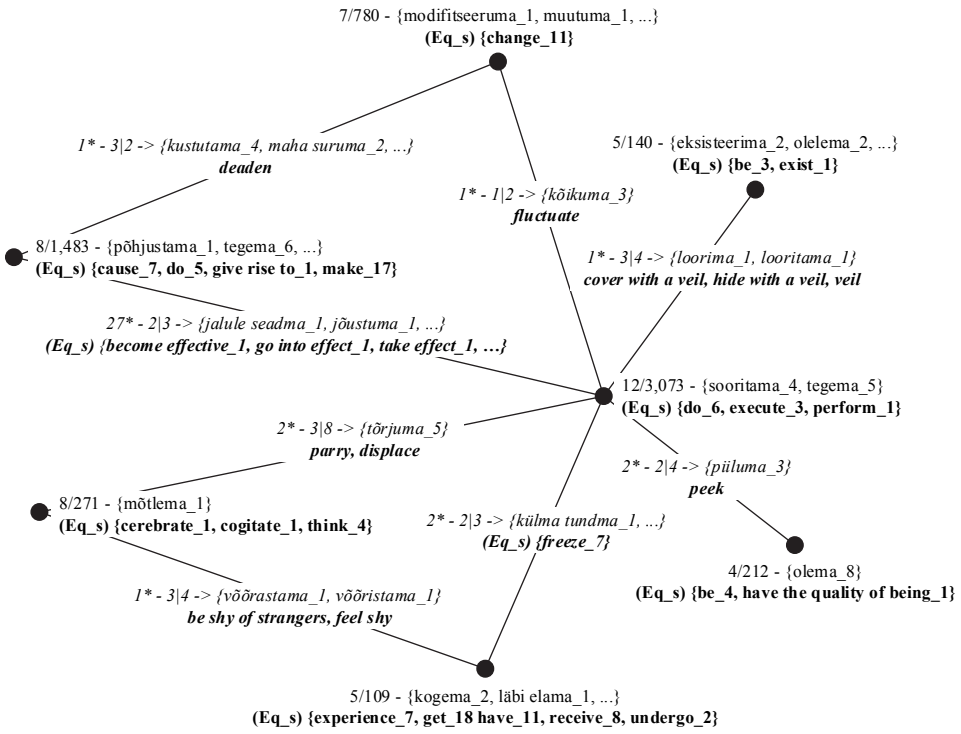


Figure 4.12. *Connected verb roots, EstWN (version 65)*

A second example of the same pattern in Figure 4.12 is about the EstWN verb hierarchy. This image depicts seven interconnected roots. Here, only concepts which have been corrected in the latest version of EstWN are considered. Firstly, roots are discussed followed by the concepts on the edges.

Presently, the latest EstWN version has only four separate (independent) verb roots. All of them are also depicted in Figure 4.12 – {change_11}, {be_3, exist_1}, {do_6, execute_3, perform_1} and {be_4, have the quality of being_1}. The remaining three roots are organized under two roots {be_3, exist_1} and {do_6, execute_3, perform_1}. Each concept as a label on the edge now only has one direct parent instead of two. Five of them flow into root {do_6, execute_3, perform_1}, two of them go to {be_4, have the quality of being_1} and the remaining one falls into {change_11}.

Below is an overview of the changed concepts shown as a list of chains according to the latest version of EstWN. The first concept in the chain is the concept from Figure 4.12 which is later changed and the last one is the root concept.

Changed roots

- {cause_7, do_5, ...} → {do_6, execute_3, ...}
- {cerebrate_1, cognitive_1, ...} → {cause_7, do_5, ...} → {do_6, execute_3, ...}
- {experience_7, get_18, ...} → {exist_2, live_4, ...} → {be_3, exist_1}

Changed concepts on the edges

- "cover with a veil" → {dress_7, get dressed_1} → {locate_3, place_26, site 3} → {act_12, do something_1, ...} → {do_6, execute_3, ...}
- "peek" → "appear, seem" → {be_4, have the quality of being_1}
- {freeze_7} → {feel_12, perceive_1, ...} → {experience_7, get_18, ...} → {exist_2, live_4, ...} → {be_3, exist_1}
- "be shy of strangers, feel shy" → "be shy, be afraid of something"→ {consider_1, reckon_3, ...} → {believe_3, think_6} → {cerebrate_1, cognitive_1, ...} → {cause_7, do_5, ...} → {do_6, execute_3, ...}
- "parry, displace" → {attitudinise_1, attitudinize_1} → {believe_3, think_6} → {cerebrate_1, cognitive_1, ...} → {cause_7, do_5, ...} → {do_6, execute_3, ...}
- {become effective_1, go into effect_1, ...} → "start, begin, set about (doing)" → {approach_12, deal with_4, ...} → {begin_4, start_19} → {act_12, do something_1, ...} → {do_6, execute_3, ...}
- "deaden" → {destroy_3, ruin_6} → {cause_7, do_5, ...} → {do_6, execute_3, perform_1}
- "fluctuate" → {change_11}

4.2 THE CASE STUDY OF A DENSE COMPONENT

This section is partially based on the paper "Dense component in the structure of wordnet" (Lohk et al., 2014a). This paper gave an overview of the inconsistencies which the test pattern helps to detect. Particular focus is on all the various corrections made by the lexicographer. The authors of this paper find that the greatest benefit of using instances of *dense component* is their help in detecting whether the *multiple inheritance* cases are justified. An in-depth analysis of the Estonian Wordnet Version 66 was performed. Some comparative figures are also given for the Estonian Wordnet (EstWN) Version 67. In the analysis of hierarchies, only *hypernymy* relations are used. The number of dense component instances in Version 66 diminished after correction from 121 to 24 in Version 67.

4.2.1 The number of multiple inheritances

The correction of a *dense component* affects the number of *multiple inheritance* cases. Looking at Table 4.1, it is clear that after the correction of *dense component* instances there are no *synsets* with 5 parents in Version 67. *Synsets* with 3 parents are reduced by about 50% and dual inheritance is reduced in about 500 cases.

Table 4.1 *The multiple inheritance counts before and after the correction of dense components.*

Nr of Parents	**EstWN, v66** (number of *synsets*)	**EstWN, v67** (number of *synsets*)
5	1	–
4	5	1
3	68	32
2	1,603	1,131
SUM	1,677	1,164

4.2.2 Distribution of dense component instances corrections

Table 4.2 gives a detailed overview of the corrections made by the lexicographer. This table is based on a manual comparison of the instances of *dense component* from EstWN Version 66 to Version 67. The sum of the first column numbers (106+14+65+39+14) in Table 4.2 is not equal to 121 because many types of corrections have been included in the same instances.

The figure 106 in the first row indicates that a *dense component* as a pattern is particularly useful in checking the jusitifications for the *regular polysemy* cases. If *regular polysemy* is not justified, it means that some semantic relations have just been removed. Due to background *synsets* that were added to every *dense component* instance (represented by dotted lines), it appears that *the principle of the economy was not followed* in the second row.

While asymmetric ring topology is possible where a direct link exceeds/overpasses more than one level of the hierarchy, an instance of *dense component* cannot be expected to refer to all of these inconsistencies.

Table 4.2 *The distribution of instances of the dense component corrections*

106	Regularity of *multiple inheritance* was not justified	
14	The principle of economy was not followed	
65	Dense components were connected to changes in semantic relation	
	162	*Semantic relationship was changed to*
		88 *Near synonymy*
		52 *Fuzzynymy*
		20 *Holonymy*
		2 *Meronymy*
39	Hierarchy was changed in the cases of	
	14	*Co-hypernyms/co-hyponyms, one became parents to another one*
	7	*Connection to a synset is replaced with another one*
	5	*New synsets were added*
	4	*Added or removed lexical units from synsets*
	3	*Synsets were merged*
	2	*Removed synsets*
	4	*Hierarchy structure was reorganized*
14	No correction needed	

Only 14 instances did not require any corrections. However, Version 67 consists of 24 instances. Their content was as follows:

- 14 of them were without any correction
- 2 of them were changed slightly
- 8 of them were new

Furthermore, all the instances of *dense component* in Version 66 were revised, and 1,868 *synsets* and 1,181 semantic relations added. Therefore, 8 new instances are included in Version 67.

4.3 CONCLUSIONS

In this chapter, the instances of all test patterns were discussed. Chiefly, instances of EstWN were used, but two examples were about PrWN. Every example was validated, using the knowledge of thesis author. Also, each example was compared with the latest version of wordnet in a web application. In the case of a *root in closed subset*, even a special web page for the ranks in the armed forces was used. However, in everyday practice the expert linguist/lexicographer validates the semantic network of wordnet. The validating process itself may be conducted at any time and for many reasons, depending on the individual development process. Nevertheless, some of the reasons for validating might be as follows:

- Checking the quality of a new release of wordnet before it will be made publicly available
- Checking the changes in a wordnet semantic network after the new concepts and semantic relations are added (including using some different or new approach for semantic network expanding)
- Checking the work of the lexicographer responsible for semantic network expanding

The second section studied a case study of the *dense component*, which was presented in the results of our paper (Lohk et al., 2014a). Comparing the instances of *dense component* in two sequential versions (66 and 67) of EstWN, we found that even though the lexicographer only corrected 107 instances out of 121, the number of *multiple inheritance* cases were reduced in 513 cases. This aspect indicates that the impact of the *dense component* to *multiple inheritance* in the validating of the semantic hierarchies of wordnet is great.

Another essential observation was new instances that came forward in the new version 67. That confirms the constant need to validate the semantic network of wordnet. That is to say, the validation of wordnet content is an infinite iterative process.

Finally, based on the example instances in this chapter, we may claim that an instance of the test pattern may help to discover a lot of errors (Sections 4.2.4 and 4.2.9), even atypical for certain test pattern instances.

5. PROGRAMS AND THE RESULTS OF THEIR APPLICATION

"The larger the network (wordnet) is, the more difficult is to keep it consistent and to minimize the number of errors in it." –
Maciej Piasecki • Łukasz Burdka • Marek Maziarz

At the beginning of our studies, the hope was to create a simple and user-friendly programs for every test pattern that could be used in wordnets for different languages. We developed algorithms and created programs to automatically find instances of the different types of test patterns. However, we implemented some algorithms and programs to semi-automatically find instances of different types of test patterns, such as *closed subset* as a bipartite graph including *the largest closed subset* and *connected roots*. In this chapter, we describe the main actions of our programs and apply them to different language wordnets to **check** their semantic hierarchies. That is to say, with the help of our programs we verify the existence and number of test pattern instances in the semantic hierarchies of wordnet numerically.

This chapter is divided into four sections. Firstly, five wordnets used for finding instances of test patterns are described. Furthermore, the main actions of the programs that were created for each test pattern are explained.

The second section provides an overview of EstWN's iterative evolution. The impact of the use of test patterns on the semantic structure of EstWN is considered from versions 60 to 70. Moreover, we were surprised by the high number of corrections made to the *synsets* and *hypernymy* relations across these 10 EstWN versions when the test patterns were applied.

The third section gives a numerical overview of the test pattern instances in four other wordnets – Princeton WordNet, Finnish Wordnet, Dutch Wordnet and Polish Wordnet.

A summary of all the main results considered in this chapter is in the fourth section, alongside proposed future work.

This chapter is based mainly on unpublished results. Only the description of the wordnets (Section 5.1.1) and some test patterns' instances numbers (Table 5.9 Wordnets in comparison) originate from the paper "Some Structural Tests for Wordnet with Results" (Lohk et al., 2014c).

5.1 WORDNETS AND PROGRAMS

In order to test our programs, wordnets from five different languages were chosen – Estonian (Estonian Wordnet), English (Princeton WordNet), Finnish (FinnWordNet), Dutch (Cornetto) and Polish (plWordNet). Disregarding the distinctions in their development, they are described thusly:

- Estonian Wordnet is chosen due to the fact that it is in our own language.
- Princeton WordNet is the first wordnet in the world, the most popular one, the "mother" of all wordnets and it is the most referred to and studied one.
- Finnish WordNet is a semantic hierarchy copy of Princeton WordNet. All *synsets* were translated by professional translators and semantic relations were taken over automatically (Lindén and Niemi, 2014). At the moment, it is larger than Princeton WordNet.
- Dutch Wordnet is the most expensive wordnet in the world. A licence for commercial use costs 15,000 euros[48].
- Polish Wordnet naturally keeps quickly growing, their project team consists of 35 members[49].

The common feature of Princeton, Finnish and Polish wordnets is that they are the largest wordnets in the world.

5.1.1 Description of wordnets

Princeton WordNet (PrWN)

WordNet (Fellbaum, 1998a) has become one of the de-facto standard lexical resources in Natural Language Processing (Farreres et al., 1998). PrWN is a large manually constructed semantic network. It was composed by a team of expert linguists and psycholinguists headed by George A. Miller at Princeton University's Cognitive Science Laboratory in 1985. After the death of G. A. Miller in 2007, the team leader is C. D. Fellbaum. PrWN has become the "mother" to all wordnets (Fellbaum, 1998b) and is undoubtedly the most referred to and studied wordnet in the world. The importance of PrWN is also due to the fact that according to the webpage of *The Global WordNet Association*[50] all wordnets (more than 70) referred to in there include links to PrWN or other wordnets that are linked to PrWN. Version 3.1 of PrWN consists of 117,773 *synsets*, including 206,779 *lexical units*.

[48] http://www.cltl.nl/projects/previous-projects/cornetto/
[49] http://plwordnet.pwr.wroc.pl/wordnet/team
[50] http://globalwordnet.org/wordnets-in-the-world/

Estonian Wordnet (EstWN)

The Estonian Wordnet began as a part of the EuroWordNet project (Vossen, 1998b) 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. After the beginning, several methods have been used, for example domain-specific ones, 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*.

Polish Wordnet – plWordNet (PlWN)

Work on PlWordNet began in 2005 (Derwojedowa et al., 2008). Its developers decided not to translate lexical concepts from PrWN trees because these trees reflect the structure of English rather than Polish. Thus, they built the semantic network from scratch. PlWN development was organized in an incremental manner, starting with general and frequently used vocabulary. The most frequent words from a reference corpus of the Polish language were selected. Version 2.0 of PlWN consists of 116,319 *synsets*, including 160,169 *lexical units*.

Cornetto (CorWN)

The goal of Cornetto was to build a lexical semantic database for Dutch, following the structure and content of Wordnet and FrameNet. Cornetto comprises of information from two electronic dictionaries: the Referentie Bestand Nederlands, which contains FrameNet-like structures, and the Dutch wordnet (DWN), which utilizes typical wordnet structures. DWN has a similar structure to the English WordNet, although the top-level hierarchy was developed from an ontological framework and more horizontal relations are defined. The database has 70,371 *synsets* and 119,108 *lexical units*.

Finnish Wordnet – FinnWordNet (FiWN)

The Finnish Wordnet project started about in 2010. Professional translators directly translated more than 200,000 word senses in PrWN (version 3.0) in 100 days. The direct translation approach was "*based on the assumption that most synsets in PrWN represent language-independent real-world concepts*" (Lindén and Niemi, 2014). The benefit of this wordnet product is a wordnet that is directly aligned with PrWN. As a side-effect of the evaluation, FiWN developers extended their wordnet to up to 120,449 *synsets* and 208,645-word senses in version 2.0. Thus, FiWN is statistically larger than PrWN.

5.1.2 Data conversion and database structure

Our goal was a program that works in every language wordnet. There is a proposed wordnet format in the context of the EU KYOTO project – the LMF (Lexical Markup Framework) format for wordnets (Henrich and Hinrichs, 2010). However, this format is not widely used. Therefore, there was a need to implement conversion programs for every wordnet we used for checking. Below is a brief overview of the origin of the wordnet databases and their data format.

Origin and data format of wordnet databases

- All **EstWN** versions were obtained from Kadri Vare from the University of Tartu who exported them from Polaris as structured plain text files.
- **PrWN** version 3.0 came from Neeme Kahusk from the University of Tartu as an exported plain text file of Polaris.
- **PrWN** version 3.1 was delivered by Randee I. Tengi from Princeton University as an SQL-database.
- For all **PlWN** versions, downloadable links were emailed to us after registration on the website of the developers[51]. All databases were in the XML-format.
- **FiWN** version 2.0 was downloaded from the website of developers[52] as plain text files.
- **CorWN** version was downloaded as an XML-file from the webpage of TST-Centrale[53] after registration.

In addition to having a different data representation format, each wordnet may also contain information on different tags. Moreover, it was discovered that the same semantic relations between *synsets* were denoted differently. However, the minimum information required was *synset IDs*, *synsets* (as the sets of *lexical units*), *semantic relations* between *synset* IDs and, if possible, *glosses* (*synset* definitions). A separate relational database was created with unified notations for every wordnet version. Each database consists of three tables – semantic relations (REL), *synsets* (SS) and definitions (DEF) (see Figure 5.1). It should be noted that every wordnet version database is in a separate file and has table relationships, as shown in Figure 5.1.

[51] http://nlp.pwr.wroc.pl/plwordnet/download/?lang=eng
[52] http://www.ling.helsinki.fi/en/lt/research/finnwordnet/download.shtml
[53] http://tst-centrale.org/

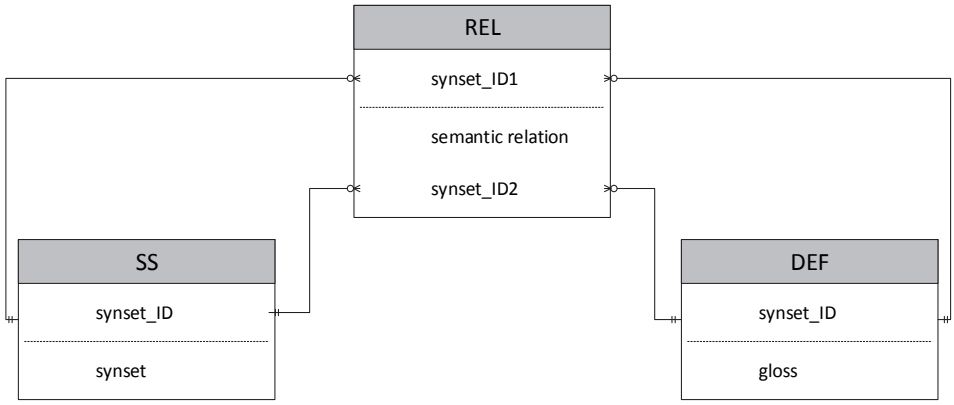


Figure 5.1. *Relationships between Wordnet database tables*

5.1.3 Main actions in the work of programs

Our programs utilize three different sequences of acts. The first is for instances that will be found automatically and the second and third ones for instances found semi-automatically. The common user interface for all programs is similar to Figure 5.2.

As regards all of the programs, the user selects a wordnet database from the combo box and clicks on the button "START". Most of the instance types will be found automatically by clicking on the "START" button, but some of them will be found semi-automatically, such as *closed subsets* which include the largest one and *closed subsets with roots* and *connected roots*.

The general functions in programs that create instances automatically are:

- all the data of the database is saved into memory arrays
- instances of the particular test pattern are detected
- all instances of certain test pattern are drawn one by one, shown to the user in user interface and then saved as a separate document with a file name that contains:
  - wordnet name and version
  - test pattern name
  - the number of instances

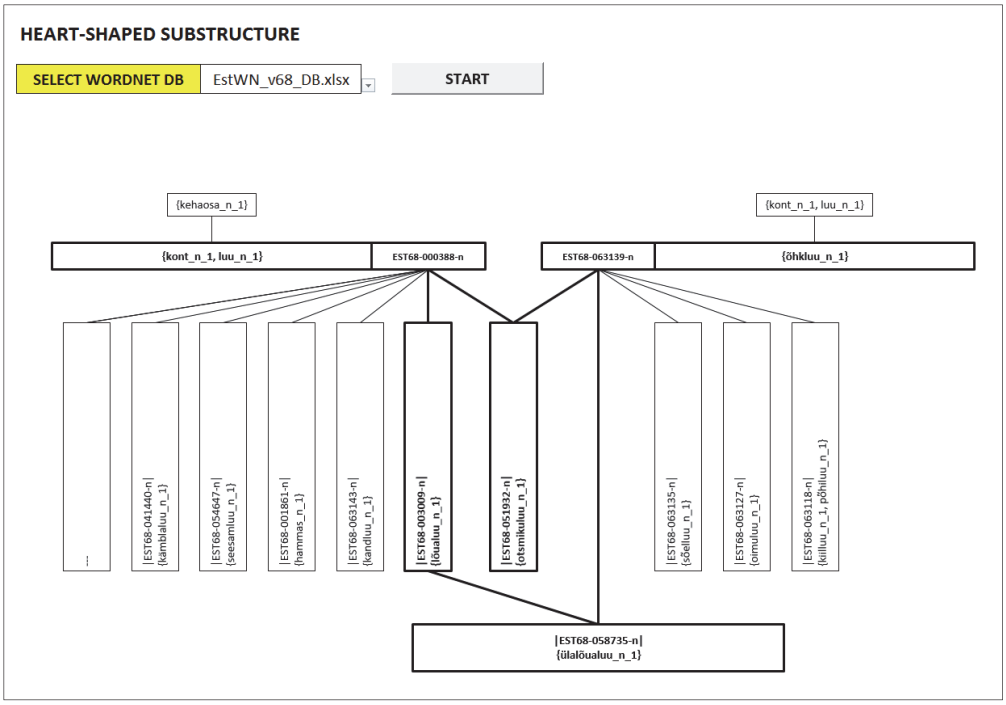


Figure 5.2. *User interface of application of the pattern of heart-shaped substructure*

The program of *connected roots*, on the other hand, automatically creates only the source file for the visualization program Pajek (De Nooy et al., 2011). To depict the instances as a graph, the user has to use Pajek and open the source file in it.

The program of *closed subsets* is only half-way developed. At the moment it

- automatically finds all *closed subsets* as interval graphs
- allows to user to select an interesting one with a double click and creates a Boolean matrix out of it
- allows the used to copy the Boolean matrix to another application that turns it into a bipartite graph with a minimized crossing number
- In the case of *large closed subset*, a fast algorithm is used to minimize the number of crossings, which was developed by Ottokar Tilk in SciLab[54] (Lohk et al., 2013).

[54] http://www.scilab.org/

5.2 AN OVERVIEW OF ESTONIAN WORDNET ITERATIVE EVOLUTION

The first attempt to check the structure of EstWN took place with version 55. One of the things studied was how many branches a *synset* goes through until it arrives at one or several *root synsets*. Presenting our results at the Estonian Applied Linguistics Conference in spring 2011, Kadri Vider[55] provided our first feedback. Her comments elucidated that EstWN is in need of that kind of structure checking.

In this section, a numerical summary of the corrections made by lexicographers in EstWN versions 60 to 70 are discussed. This summary includes corrections related to the noun and verb *lexical units* and corrections related to *hyponymy* relations. In this process, version pairs 60-61, 61-62, 62-63, 63-64, etc. up to 69-70 are compared. The older version is taken for granted in this comparison. It means that noun and verb *lexical units* are taken from an earlier version and compared to a newer one. This is also done with *hypernymy* relations.

Secondly, we look at how the number of test pattern instances changes in the numerous versions of EstWN.

5.2.1 Correcting statistics of EstWN

As shown by the subsequent statistics, during its development process, EstWN also goes through the process of correction. Our statistical overview only includes changes in noun and verb *synsets* and *hypernymy* relationships between two sequential wordnet versions. While *synset* ID may vary in different versions, we took *lexical units* **(LUs)** with their sense numbers for granted. It is important to note that the statistics do not consider cases of new *synsets* and their new semantic relationships**.**

Our statistics is divided into two parts – statistics related to *synsets* and that related to *hypernymy* relations. We represent both parts separately.

Synset statistics

There are three separate features considered in the *synset* statistics (Table 5.3). *Synsets* which are removed because all their members were removed belong into the **first group**. For example, a comparison of versions 60 and 61 shows:

> {pakend_1} (package) exists in version 60 but is removed in version 61.
>
> (It is interesting to note that "pakend_1" appears again in version 62 together with other *lexical units* and from version 64 onwards, it is represented again as it was in version 60 – {pakend_1}.)

[55] from the University of Tartu

The second group contains *synsets* where at least one *lexical unit* is removed, but the *synset* itself is preserved. For example, comparing versions 66 and 67:

{tüdruk_1, neid_2, tütarlaps_1} (girl, maiden, young girl)
→ removed: "tütarlaps_1" (young girl)
→ result: {tüdruk_1, neid_2} (Eq_s){child_5, female child_1 girl_2, little girl_1}

The third group encompasses *synsets* which have been merged, divided into many *synsets* or have at least one new *lexical unit*.

Merged *synsets* (comparing versions 66 and 67)
{abielumees_1} (married man)
{mees_3, abikaasa_2} (hubby, husband)
→ {mees_3, abikaasa_2, abielumees_1}
(Eq_s) {hubby_1, husband_1, married man_1}

Synsets divided into many *synsets* (comparing versions 65 and 66)
{aisting_1, mulje_2}
→ {aisting_1} (Eq_s) {sensation_1, sense datum _1, ...}
→ {mulje_1} (Eq_s) {belief_1, feeling_5, impression_4, notion_3}

A new *lexical unit* added to the *synset* (comparing versions 68 and 69)
{magistriõpe_1} (master's studies)
New *lexical unit* "magistratuur" (master's course)
→ {magistriõpe_1, magistratuur_1} (master's studies, master's course)

Table 5.3 *Synset statistics*

Compared versions	I-group. Removed LUs with removed *synsets*		II-group. Removed LUs with changed *synsets*		III-group. Changed *synsets*
	Lexical units	*Synsets*	*Lexical units*	*Synsets*	
60_61	683	575	36	31	462
61_62	502	399	162	139	503
62_63	223	219	17	17	146
63_64	30	21	40	33	442
64_65	41	36	111	105	912
65_66	333	271	50	43	566
66_67	46	35	50	43	480
67_68	16	12	18	17	398
68_69	39	31	14	14	393
69_70	38	26	14	11	589
SUM	1,951	1,625	512	453	4,891

Statistics of hypernymy relations

In *hypernymy* relation statistics, two types of corrections are distinguished – removing and replacing. At the moment, the removal operations were not counted where a *synset* was removed (Table 5.4). Upon comparing Table 5.4 with Table 5.5, it is clear that the removal operation finds much more application. One of the reasons is that there are hypernyms (concepts) that group together a number of direct subconcepts but which have the wrong connection basis to their superordinates.

Table 5.4 *Statistics of removing hyponymy and hypernymy relations*

Compared versions	*Hyponymy* and *hypernymy* relations removed	The most frequently used *hypernym* in the removal of *hyponymy*-relations of its subordinates	Frequency
60_61	2,299	{teadus_1, teadusala_1, ...} **(science, science discipline)**	198
61_62	1,898	{elund_1, organ_2} (Eq_s)**{organ_4}**	48
62_63	470	{firma_1} (Eq_s)**{business firm_1, firm_1, ...}**	213
63_64	1,342	{haigus_1, tõbi_1} (Eq_s)**{disease_1}**	293
64_65	3,052	{kast_1} **(social rank, social station, social status)**	56
65_66	2,812	{liikuma_3} (Eq_s) **{change position_1, move_14}**	39
66_67	3,054	{muutma_2} **(alter, change)**	80
67_68	1,951	{päev_4} (Eq_s)**{day_4 mean solar day_1, ...}**	49
68_69	2,266	{sooritama_4, ...} (Eq_s) **{do_6, execute_3, perform_1}**	48
69_70	2,767	{tegevus_1, toiming_2, ...} (Eq_s) **{activity_1}**	119
SUM	21,911		1,143

The last column in Table 5.4 provides better context to the second column of statistics. It is a notable fact that there are *hypernyms* which have taken part in removing several relationships. The most frequently used ones are represented in the fourth column. An interesting fact is that some *synsets* were handled to over 200 times in such corrections (Table 5.4).

The next example (comparing versions 60 and 61) represents a case where a *hyponymy* relation is apparently removed due to the fact that "cigarette" is a too specific concept for "artefact". Later, "cigarette" is given the parent "tobacco products".

{artefakt_n_2, asi_n_4, tehisasi_n_2} (Eq_s) {artefact_1, artefact_1} →
hyponymy →
{suits_2, sigaret_1} (Eq_s) {cigarette_1, cigarette_1, fag_1}

Table 5.5 *Statistics of replacing hyponymy/troponymy and hypernymy relations*

Compared versions	*Hyponymy* and *hypernymy* relation replaced	The most frequently used *hypernym* in the replacement of *hyponymy*-relations of its subordinates	Frequency
60_61	74	{hotellikett_n_1} **(hotel chain)**	2
61_62	148	{korruselamu_n_1} **(multi-storey building)**	4
62_63	98	{käigukang_n_1, käigukast_n_1} (Eq_s) {**gear case_1, gearbox_1**}	2
63_64	120	{kõneakt_n_1, suhtlusakt_n_1} (Eq_s) {**speech act_1**}	3
64_65	814	{ehitustööriist_n_1, remonditööriist_n_1,...} **(building tool, repair tool)**	76
65_66	222	{kõneakt_n_1, suhtlusakt_n_1, ...} (Eq_s) {**speech act_1**}	3
66_67	816	{filosoof_n_1, mõttetark_n_1} (Eq_s) {**philosopher_1**}	114
67_68	854	{hügieenitarbed_n_1} **(hygiene utensils)**	4
68_69	660	{kõneakt_n_1, suhtlusakt_n_1, ...} (Eq_s) {**speech act_1**}	3
69_70	316	{loomasaadus_n_1} **(animal product)**	4
SUM	4,122		215

In Table 5.5 we see how many times *hypernymy* relations are replaced with new semantic relations. This correction operation took place 4,122 times. The most common alternative relation to *hyponymy* or *hypernymy* was *near_synonymy*, which occurred about 1,800 times. This was followed by *fuzzynym*, at about 800 times. Secondly, in the operation of *replaced hypernymy relation*, the frequency number of *synsets* which handled this correction most frequently are much lower comparing to

previous correction numbers in Table 5.4. The *synset* {philosopher} represented an exception, which took part 114 times in the corrections of *replaced hypernymy relation.*

In addition to the different corrections for *synsets* and semantic relations, it is worth mentioning that there are about 950 *lexical units* among nouns and verbs for which *synonyms* are changed at least twice. An example is "toit_1" together with "roog_2" and "söök_2" from the same *synset* – (Eq_s) {dish_3} *(a particular item of prepared food)* – their *synonyms* have changed four times, as shown in Table 5.6.

Table 5.6 *A synset in the change process*

Version	*Synsets*
63	{**roog_n_2**, **söök_n_2**, **toit_n_1**}
64	{pala_n_3, **roog_n_2**, **söök_n_2**, **toit_n_1**}
66	{pala_n_3, **roog_n_2**, söögilaud_n_4, **söök_n_2**, toiduaine_n_2, toidulaud_n_3, **toit_n_1**}
67	{pala_n_3, **roog_n_2**, **söök_n_2**, **toit_n_1**}
68	{leivapoolis_n_2, pala_n_3, **roog_n_2**, söögipoolis_n_2, **söök_n_2**, toidupoolis_n_2, **toit_n_1**}

5.2.2 The use of test patterns

In order to employ the *closed subset* patterns in EstWN (version 60), a collaboration was started with linguists-lexicographers Kadri Vare and Heli Orav from the University of Tartu (Lohk et al., 2012a), (Lohk et al., 2012b), (Lohk et al., 2014a), (Lohk et al., 2014b). This collaboration was beneficial to both sides – the lexicographers were interested in validating and correcting their wordnet and we were interested in finding out how useful the test patterns are in the validation process.

Table 5.7 reflects the release date of different EstWN versions together with the adaptation of test patterns. The latter shows when test patterns were taken into use or in which EstWN version instances of test patterns were sent to lexicographers. As illustrated by the table below, the *closed subset* test pattern is not much used. This is because the instances of *closed subset* can often be too large for convenient handling. In addition, finding instances of *closed subset* are still semi-automatic.

Table 5.7 *The release dates of the EstWN versions and test pattern adaptations*

Releasing date	version	Closed subsets	Short cuts	Heart-shaped substructure	Dense component	„Compound" pattern[56]	Multiple inheritance cases*	Rings	Synset with many roots
April, 2011	60	x							
September, 2011	61	x							
November, 2011	62	x							
January, 2012	63		x						
April, 2012	64		x						
September, 2012	65		x	x					
January, 2013	66		x		x				
May, 2013	67		x	x	x	x	x		
September, 2013	68		x	x	x	x	x	x	
December, 2013	69		x	x				x	
July, 2014	70		x	x	x	x	x	x	
December, 2014	71		x	x	x	x	x	x	x

There are three test patterns that were never sent to a lexicographer to be analysed – *large closed subset*, *connected roots* and *roots in closed subset*. However, they have been introduced at different conferences (Section 3.3).

5.2.3 A numerical overview

As shown in Table 5.7, different types of test pattern were applied at different times and versions. It is important to note that every time a lexicographers add a new *lexical unit* or *synset*, they may also correct the semantic network of wordnet. Furthermore, every new wordnet version brings along new instances of test patterns. For example, the *short cuts* are the only pattern that requires 100% correction. Nevertheless, after the correction of that type of instances, the newer version still reveals them again. Moreover, together with the test pattern, *root synsets* are delivered to the lexicographers in a random way with information about their depth and the number

[56] Substructure that considers the content of synsets

of subordinates. In addition, there is information about "orphan" nodes (*null graphs*) – *synset* without any semantic relation.

Table 5.8 *A numerical overview of EstWN spanning eleven versions*

Version	Noun roots	Verb roots	Multiple inheritance cases	Short cuts	Rings	Synset with many roots	Heart-shaped substructure	Dense component	„Compound" pattern	The largest closed subset
60	142	24	1,296	235	3,445	1,123	1,825	104	301	3,057×457
61	183	22	1,592	259	3,560	1,309	1,861	121	380	3,344×472
62	102	16	1,700	299	3,777	1,084	1,941	128	415	2,970×356
63	114	16	1,815	321	3,831	1,137	2,103	141	447	4,103×405
64	149	15	1,893	337	3,882	1,173	2,232	149	471	4,374×425
65	248	14	1,717	194	2,171	791	451	132	459	3,875×263
66	144	4	1,677	119	1,796	613	259	121	671	2,907×218
67	129	4	1,164	79	928	477	167	24	407	319×21
68	131	4	691	60	537	232	38	18	54	319×21
69	121	4	102	18	291	35	1	8	23	350×7
70	118	4	51	7	21	70	0	3	7	123×4

Based on Table 5.8, since versions 65 and 66, almost all the numbers of instances have started to decrease. A particularly high impact on the reduction appears in version 68, where many other test patterns were used in addition to all the *multiple inheritance* cases. Beside information about the number of test pattern instances,

Table 5.8 contains information about the number of noun and verb *root synsets* as well as *multiple inheritance* cases. The first two help to validate wordnet hierarchies globally. Despite the fact that the numbers of noun roots are too big (Section 2.2.1), about 75% (in version 70, 88 of 118) of roots are related to hierarchies with one additional level. **These cases prove that the correction of EstWN is not complete yet.**

5.3 DIFFERENT WORDNETS IN COMPARISON

This section provides a few comments about four wordnets which are represented in numbers in Table 5.9. This table has the same fields as Table 5.8 for 11 EstWN versions – the instances are only shown for automatically detectable test patterns. The only difference in comparison to Table 5.8 is in the field of *the largest closed subsets* (LGSs), where the first five largest *closed subsets* are shown.

As FinWN was translated on the basis of PrWN Version 3.0, it is interesting to compare these both (Table 5.9). In spite of the fact that FinWN has been expanded and corrected in version 2.0, it still has quite similar numbers of instances in comparison to PrWN Version 3.0. Even *the largest closed subsets* are rather analogous. Furthermore, the largest verb *closed subsets* are the same size and in the same 50th position. When comparing PrWN Versions 3.0 and 3.1, it must be agreed that the semantic network of the latter version has not changed significantly. Nevertheless, PrWN and FinWN differ from other ones due to the fact that LGSs are from a noun hierarchy and the first verb *closed subset* is far from the first position.

Table 5.9 *Wordnets in comparison*

Version	Roots - noun	Roots - verb	Multiple inheritance cases	Short cuts	Rings	Synset with many roots	Heart-shaped substructure	Dense component	„Compound" pattern	The largest closed subsets
PrWN v3.0	12	334	1,453	40	2,991	18	155	115	358	1,333×167-n 377×34-n 143×27-n 504×24-n 141×19-n 50. 60×4-v
PrWN v3.1	12	340	1,425	41	2,821	21	149	107	366	1,064×126-n 366×33-n 143×27-n 152×26-n 500×24-n 52. 60×4-v
FinWN v2.0	12	334	1,453	40	2,991	18	155	115	394	1,334×167-n 409×34-n 143×27-n 527×24-n 143×19-n 50. 60×4-v
CorWN v2.0	2	2	2,438	351	5,309	62	1,226	217	549	11,032×589-n 4,423×545-v 317×43-n 233×17-n 62×8-v
PlWN v1.8	669	44	10,155	546	36,670	118,466	4,894	734	672	29,638×4,321-n 3632×545-v 255×28-v 88×22-v 156×21-v

Version	Roots - noun	Roots - verb	Multiple inheritance cases	Short cuts	Rings	Synset with many roots	Heartshaped substructure	Dense component	„Compound" pattern	The largest closed subsets
PlWN v2.0	637	42	10,942	553	57,887	205,254	5,037	778	541	30,794×4,683-n 3718×551-v 393×58-v 254×28-v 88×22-v

Although CorWN is surprising in terms of the small number of noun and verb roots, its second *closed subset* is larger than the second one in both of the PlWN versions. Meanwhile, the 5th LGS of CorWN is much smaller than the ones in other wordnets.

Quite a large number of instances of *synsets with many roots* and *the largest closed subsets* in PlWN indicate two types of structure specialities. The first refers to the situation where *multiple inheritance* cases run over many hierarchy levels (vertically). This includes also all *ring*'s instances. The second one embodies the case where several *multiple inheritance* cases occur on a certain hierarchy level (horizontally).

5.4 CONCLUSIONS

Verifying the existence of test patterns' instances is one of the possible methods to measure the condition of the semantic hierarchies of a wordnet. In this work, we may take for granted the results of the last validated EstWN version. That is to say, we may use the numbers of test patterns instances from the last EstWN Version 70 in Table 5.8 and compare these numbers with the numbers of other wordnet test patterns' instances from Table 5.9. However, a second possible method to measure the condition of the semantic hierarchies of a wordnet is to compare two sequential versions of this wordnet and estimate the changes of semantic hierarchies through the numbers of instances.

In this chapter, we found the number of every test pattern's instances for five different language wordnets – EstWN, PrWN, FinWN, CorWN and PlWN. Moreover, while the use of test patterns finds application in 11 versions of EstWN, we found these numbers for these versions (from 60 to 70). In such a manner, we obtain an overview of the EstWN iterative evolution. But apart from that, we also detected the correction operations and their numbers for every pair of two sequential versions (Table 5.3 – 5.5). Here we have to consider that every EstWN version always contains correction operations that are made due to adding new *lexical unit* or *synset*

into the semantic network as well as corrections made due to applying test patterns to the semantic network.

We discovered, if we do not consider the totally new *synsets* every new version acquires, that there are quite a large number of different correction operations per EstWN version made by the lexicographer. We found that the average number of corrections per EstWN version is approximately 3,300[57] and the most frequently used correction across the 10 EstWN versions was the removal of *hypernymy* relations. This correction operation was applied 21,911 times. In that case, an average lexicographer carried out 2,200 corrections for each version. Cases where semantic relations were removed along with a *synset* were not considered.

The first test pattern instances we delivered to lexicographers for EstWN version 60 was the *closed subset* (Table 5.7). We used this pattern for some experiments in our first two joint papers. Later we brought *short cut* into use and from version 65 *heart-shaped substructure* and since version 66 *dense component*. The bigger changes in the number of *multiple inheritance* cases began with versions 65 and 66 (Table 5.8). For example, from version 66 to version 70, the number of EstWN *multiple inheritance* cases (thus also the number of test patterns' instances) in the semantic hierarchies of IS-A relation are reduced approximately 97%.

In the future, because we have all the information about the corrections related to *synsets* and *hypernymy* relations made by a lexicographer in versions 61 to 70, we believe that this will be useful for another kind of feedback. For instance, it could indicate whether changes to *synsets* and *hypernymy* relations have been conducted in a systematic way.

Finally, as the latest version (70) of EstWN in Table 5.8 reveals far smaller numbers of test pattern instances than the other four wordnets in Table 5.9, we would recommend applying these patterns to the other four wordnets as well.

[57] From tables 5.3 – 5.5: (1,625+453+4,891+21,911+4,122)/10 = 3,300.2

CONCLUSIONS AND FUTURE WORK

This chapter summarizes the main results of this thesis and proposes work that could be undertaken in the future.

Developing a semantic network of wordnet is a big challenge. **At first**, developers have to choose the appropriate approach for wordnet building – what lexical resources, building models or automation levels to use. **Secondly**, how to avoid introducing errors into the semantic network during the building and expanding process. **Thirdly**, how to keep the network constantly clear of errors or how to frequently validate the semantic network of the wordnet. In this work, we deal with this third phase. More specifically, how to check and validate the semantic hierarchies of the wordnet.

Indeed, no matter how the construction of the semantic hierarchies is carried out; there will always be possibilities for importing errors into its network. For instance, if aiming to translate source wordnet to target wordnet some of following errors may appear: translation errors, errors in source semantic hierarchies, errors related to different semantic hierarchies in both languages, and errors related to culture and region specific concepts.

DISCUSSION OF THE RESEARCH METHODS AND APPROACHES EMPLOYED

A variety of methods for validating the semantic hierarchy of a wordnet already exists. In this thesis, we classified them into three groups characterized by two features – *whether they use additional lexical resources* and *whether the content of the synset is considered*. The first group of methods – *lexical resources based* – use both of these features. The second group – *rules system based* – only use the second feature, and the third group – *graph-based* – does not use any of these features. Our approach, the *graph-based* one, differs from others by at least two aspects: in validation it uses *specific substructures in semantic hierarchies* and also gives an *overview of the threaded hierarchies and their size and depths*.

The graph-based approach is the most formal, **does not depend on the language of a semantic network**, but is up to now most rarely used in practice. Some papers that study the graph-based approach deal with substructures such as *cycles*, *null graphs*, *short cuts*, *rings*, and *dangling uplinks*. One reason for their infrequent use may be that most of them are automatically avoidable with the wordnet management system. However, **one of the main objectives of this work is to prove that in addition to *cycles, rings, dangling uplinks* and *null graphs*, there are other kinds of subgraphs which can also be helpful in validating the semantic hierarchy of a wordnet**.

We refer to these substructures, most as yet undiscovered, some inspired by other authors, as *test patterns*. Their common feature is *multiple inheritance*, which is often prone to different semantic errors. Semantic errors have a variety of causes, and they will be explored by test patterns that propose different perspectives on the semantic hierarchies of wordnet. To do this **we implemented algorithms and made programs for every test pattern** and we used them mainly to check and validate the IS-A semantic relation in different versions of Estonian Wordnet semantic hierarchies.

Our approach is based on the following sequence of actions:

- after a new wordnet version release, the lexicographer send us the new wordnet database,
- our program detects all the particular subgraphs (test pattern instances) from a certain wordnet version and saves them as a file,
- the lexicographer validates the instances and corrects them in the wordnet management system, if necessary
- In this work, we verified the existence of test patterns' instances and their numbers across 11 Estonian Wordnet versions where test patterns were used for the validation of semantic hierarchies and we discovered that the wordnet semantic hierarchies changed drastically. For example, from version 66 to version 70, the number of EstWN *multiple inheritance* cases (also the number of test patterns' instances) in the IS-A relation's semantic hierarchies decreased approximately 97%.

Although test patterns give good results in the case of Estonian Wordnet, it is important to note that *multiple inheritance* is not always wrong. One reason for that, is the possibility of reorganizing all cases where *multiple inheritance* is used. For example, "hostel" may be simultaneously a "building" as well as an "institution". However, the lexicographer may decide to organize "hostel" into two *synsets*. This would eliminate the *multiple inheritance* completely. Furthermore, there are no guidelines as to which of the two options a lexicographer should choose (Verdezoto and Vieu, 2011), and so a lexicographer must rely on his or her language perception. Moreover, different lexicographers have different language perception, and their perception may change over time (Čapek, 2012). Additionally, it is doubtful that they can be consistently systematic working with polysemic words. Considering all the above, test patterns are one option for checking how systematically these polysemic words, which cause multiple inheritance cases, are handled.

As we discussed in Chapter 3, all test patterns' instances may help to detect errors that are typical to them. However, it should be understood that "typical errors" are not always definitive of some test patterns. That is to say, we cannot claim that every test pattern brings out the same typical errors for every language wordnet because they are developed using different building methods and lexical resources. This is

because some of the typical errors related to certain test patterns are specific to the wordnet. For example, when the *heart-shaped substructure* test pattern was applied to the Princeton WordNet, Princeton University linguists detected that in most of the cases this pattern's instances referred to the case where instead of an IS-A relation the *role* or *type*-relation should be used. In that case use of *role* and *type*-relation was a problem because it is not present in Princeton WordNet. For that reason, we cannot assume that the same issue is characteristic of every wordnet. Indeed, by constantly applying the *heart-shaped substructure* on Estonian Wordnet (from version 65) the number of instances dropped to zero. According to our later statistics none of the correction operations was related to exchanging an IS-A relation for a *role* or *type*-relation.

ANSWERS TO THE RESEARCH QUESTIONS

In order to answer the main question "**How to check and validate the semantic hierarchy of a wordnet?"** all the sub-questions must be answered.

(Chapter 1) How different construction approaches may affect the semantic hierarchies of wordnet?

- What is the impact of the particular feature of *hypernymy* on the semantic hierarchy?

  According to (Miller, 1998), the hypernymy relation of a noun actually has many relations. He highlighted two of them. The first is known as IS-A or IS-A-KIND-OF relation and the second one is IS-USED-AS-A-KIND-OF relation. IS-A belongs to the "taxonomic category" and IS-USED-AS-A-KIND-OF belongs to the "functional category". For example, {chicken_2} as {bird} belongs to the "taxonomic category", but {chicken_1} as {food} belongs to the "functional category". In this example, both categories are used in different hypernymy relations. However, occasionally these categories are in one relation, e.g. {poker_1, fire hook_2, ...} as {fire iron} or presented as one synset with many parents, e.g. {written agreement_1} as {legal document} and as {agreement}.

- What is the impact of polysemy on *multiple inheritance* in the semantic hierarchy of a wordnet?

  There are typically two ways of presenting a polysemous word (lexical unit) in the wordnet semantic hierarchy – the first is to organize it into two or more synsets, whilst the second is to place it in one synset that has many parents. An example of the first is in PrWN, where {samba_2} is "dance music" and {samba_3} is "dance". The second example is {cheese_1} that is simultaneously "food" and "dairy product".

- What is the impact of *regular polysemy* on the *regularity of multiple inheritance*?

  Regular polysemy means that one word is related to at least two meanings as a variation of another word with equivalent meanings. If both polysemous words are presented (separately) in one synset, the regularity of multiple inheritance is evidenced.
- What are the three aspects every wordnet creator must consider and how do they affect the quality of the semantic hierarchies of a wordnet?

  In wordnet building, a creator must choose what lexical resource(s), building model and automation levels to use. These three aspects in different combinations have a significant impact on the quality of the semantic hierarchies. On the one hand, wordnet quality depends on the quality of the lexical resource sources, yet on the other hand, the methods used in information extraction are very important. In addition, the automation level may have an impact on target wordnet culture or region specific concepts and semantic relations.

(Chapter 2) What methods are used in the validation of the semantic hierarchies of a wordnet?

- How to systematize the methods of validating employed in semantic hierarchies?

  We discovered two features that divide the methods of validating used in the semantic hierarchies of a wordnet. These features were presented as two questions: 1) whether this method use lexical resources? 2) Whether this method consider the content of a synset?

  Thus, using these two features, three groups of methods can be distinguished:
  - *I group of methods uses lexical resources and the content of synsets. It employs methods in order to extract knowledge from lexical resources. One well-known approach is to use lexico-syntactic patterns.*
  - *II group of methods uses only the content of synsets. It applies different rule systems to synsets and semantic relations to validate the semantic hierarchy. For instance, this group encompasses the use of ontology meta-properties and top-ontology features if they are part of the wordnet.*
  - *III group of methods does not use any of these two features. These methods approach the semantic hierarchy from the perspective of a graph and verify the existence of particular substructures in it, e.g. cycles, rings, dangling uplinks.*
  - Into which group of methods does our approach belong?

    Our approach for validating the semantic hierarchies of a wordnet belongs to III group.
- What types of errors occur in wordnet?

  There are roughly three types of errors:

  1) *Syntactic (also formal or surface) errors are related to the source file structure or data presentation in it. For example, a tag is empty or a record is presented twice.*

2) *Semantic errors are related to the semantics of synsets and relationships, e.g. an inappropriate lexical unit in a synset or a wrong semantic relation.*
3) *Structural errors are related to the semantic hierarchy of the wordnet as a graph, e.g. cycles or dangling uplinks.*

– Into which group of methods does our approach belong?

Our approach catches the possible structural errors that refer to potential semantic errors.

(Chapter 3) What test patterns to use in order to check and validate the semantic hierarchies of a wordnet?

This thesis was limited to patterns which contain the multiple inheritance cases. This is mainly because multiple inheritance in wordnet semantic hierarchies is often used in an inappropriate way. For example, according to (Gangemi et al., 2001), there are many cases where multiple inheritance is not used as a conjunction of two properties.

- How to describe test patterns?

  We used graph-based presentation (mathematical model) and textual description.
- What is the most similar work to ours?

  The most similar work was by (Fischer, 1997), (Liu et al., 2004) and (Richens, 2008). They provided the idea of using short cuts and rings. The idea of "synsets with many roots" partially originates from (Richens, 2008).
- What direction to follow in validation on the basis of different test patterns?

  While this thesis introduces ten different test patterns, this question is answered in a general way. Based on the fact that every test pattern instance contains multiple inheritance cases, two general questions can be proposed for each instance: whether a certain concept can simultaneously have many parents or whether the parents of the same concept are on the same hierarchical level. The directions on every test pattern instance tend to be quite specific. For instance, for "connected roots", a total of six questions were proposed, which may refer to particular inconsistencies.
- What kinds of errors are typical to every test pattern?

  Similarly to the reasoning above, this question is answered in a general way as well. In most cases (based on Chapter 4, "Test patterns in action"), we believe that every error can be classified under two general groups of errors – a wrong or a missing semantic relation. Specifically, typical errors can be the following:
  – *a synset has a connection to a specific and a general concept,*
  – *unfinished work with a subhierarchy,*
  – *the root synset is too specific to be a root synset, it belongs to another hierarchy,*
  – *the multiple inheritance is not justified or has to be expanded*

(Chapter 4) How to validate test pattern instances in the wordnet semantic hierarchies in practice?

- What are the examples for validating the instances of test patterns?

  We presented nine examples from different Estonian Wordnet versions. Two examples belonged to Princeton WordNet (version 3.1). In the problem descriptions, we followed the recommendations given in Chapter 4.

- Who validates the semantic hierarchies of a wordnet?

  Usually an expert linguists/lexicographer. However, in the future we hope to help him/her with more concrete recommendations. At the moment, further investigation is required.

- When to validate the semantic hierarchies of a wordnet

  Generally, it depends on what kind of development process is selected. In the case of Estonian Wordnet, validation occurred after each new version release.

(Chapter 5) How to check the instances of test patterns in the wordnet semantic hierarchies?

- What are the main actions of a program implemented to find test pattern instances?

  1) *Read data from database tables and save them in memory arrays, trees, and stacks.*
  2) *Find all instances of a test pattern.*
  3) *Show them one by one to the user and save them in a file.*

- What is the effect of the test patterns used for validation on the EstWN semantic hierarchies?

  According to Table 5.8 *(Numerical overview of EstWN over eleven versions), the use of test patterns is clearly mostly manifested in two points. To begin with, when a lexicographer corrected the Estonian Wordnet (version 66) on the basis of the "dense component" pattern for the first time, its instances decreased from 121 cases to 24, but this even had a strong effect on other numbers of instances, e.g. the number of "rings" decreased by 868.*

  *Secondly, from version 67 (*Table 5.7*) onwards, the lexicographer has utilized more test patterns for the validation, and other strong effects on the numbers of test pattern instances have occurred.*

  Finally, if versions 66 and 70 are compared, the number of EstWN multiple inheritance cases (thus also the number of test patterns' instances) in the semantic hierarchies of IS-A relation are reduced approximately 97%.

CONTRIBUTIONS

The main contributions of this thesis:

- **Mathematical models** as graphs (test patterns) which describe yet undiscovered substructures useful to check and validate the semantical hierarchies of wordnet-type dictionaries. In addition, these test patterns are presented along with typical errors they may help to detect and usage examples.
- **Cross-language test patterns** applicable in all wordnets in the world (there are about 50 different language versions of wordnets)
- **Test patterns as the tool of lexicographer** to check and validate the semantic hierarchies of wordnet. Test patterns simplify the work of lexicographer significantly as their instances help to point exactly to the places in the semantic hierarchies that may reveal certain type of errors.
- **Implemented algorithms** to find the instances of test patterns.
- An overview of 10 versions of **Estonian Wordnet iterative evolution** where test patterns were employed and which prove the efficiency of test patterns in validation of the semantic hierarchies of wordnet.

FUTURE WORKS

This thesis concentrated on *hypernymy* relations among noun and verb hierarchies. Nevertheless, the proposed test patterns are applicable to other types of semantic relations. In the beginning of our study, test pattern of *closed subset* were applied to *near synonymy* relations (Lohk et al., 2012a). As certain test patterns help to investigate how different concept subgroups are related, it would be interesting to see if there is any regularity in the use of *near synonymy* relations. In section 5.2, we found that the *hyponymy/hypernymy* relation was replaced with *near synonymy* in about 1,800 times. Furthermore, since the number of *near synonymy* relations is quite big (13,528) in the latest EstWN version, this semantic relation appears worth further investigation. Moreover, the *hyponymy/hypernymy* relation was replaced with fuzzynymy relation in about 800 times. This relation appears 19,136 times in the latest EstWN version. Therefore, we believe it would be fruitful to investigate this relationship as well.

Secondly, section 2.1.2 introduced a method (Nadig et al., 2008) that uses information from an explanatory dictionary to check and validate the semantic hierarchies of a wordnet. This approach presumed that every entry (head word) in the dictionary contains either its *hypernym* or *synonym*. This idea helps to construct a hierarchy to which our test patterns can be applied. For instance, the *Estonian explanatory dictionary*[58] could be used for this test.

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

Thirdly, it has been mentioned that instances of different test patterns may overlap. It means that the same *multiple inheritance* case may appear in more than one different test pattern instance. In the future, it could be useful to create an interactive application that allows the user to change the current *test pattern view*. This assists in more adequately validating particular *multiple inheritance* cases. In one of our experiments, the instances of a *dense component* had the best overlap with other test pattern instances.

Fourthly, there are about 70 wordnets in the world. All of our test patterns are applicable to them. The minimum information required is the *synsets*, their ID and the semantic relations between the *synsets*.

Lastly, the information gathered about operations made in correcting the hierarchies of EstWN spanning 10 versions can also be used. If we find a way to categorize it, this information will be another source feedback for the lexicographer.

REFERENCES

Agerri, R., García-Serrano, A., 2010. Q-WordNet: Extracting Polarity from WordNet Senses., in: Proceedings of the 7th Language Resources and Evaluation Conference. European Language Resources Association (ELRA), Malta, pp. 2300–2305.

Aliabadi, P., Ahmadi, M.S., Salavati, S., Esmaili, K.S., 2014. Towards Building KurdNet, the Kurdish WordNet, in: Proceedings of the 7th Global WordNet Conference. University of Tartu Press, Tartu, Estonia, pp. 1–6.

Alvez, J., Atserias, J., Carrera, J., Climent, S., Laparra, E., Oliver, A., Rigau, G., 2008a. Complete and Consistent Annotation of WordNet using the Top Concept Ontology, in: Proceedings of the 6th International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), Marrakech, Morocco, pp. 1529–1534.

Alvez, J., Atserias, J., Carrera, J., Climent, S., Oliver, A., Rigau, G., 2008b. Consistent Annotation of EuroWordNet with the Top Concept Ontology, in: Proceedings of the 4th Global WordNet Conference. Szeged, Hungary, pp. 1–20.

Apresjan, Y., 1974. Regular Polysemy. Linguistics 142, 5–32.

Atserias, J., Climent, S., Moré, J., Rigau, G., 2005. A Proposal for a Shallow Ontologization of Wordnet. Procesamiento del Lenguaje Natural 35, 161–167.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z., 2007. DBpedia: A Nucleus for a Web of Open Data, in: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (Eds.), The Semantic Web, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 722–735.

Azarova, I., Mitrofanova, O., Sinopalnikova, A., Yavorskaya, M., Oparin, I., 2002. Russnet: Building a Lexical Database for the Russian Language, in: Proceedings of Workshop on Wordnet Structures and Standardisation and How This Affect Wordnet Applications and Evaluation. European Language Resources Association (ELRA), Las Palmas, Canary Islands, Spain, pp. 60–64.

Baker, C.F., Fellbaum, C., 2009. WordNet and FrameNet as Complementary Resources for Annotation, in: Proceedings of the Third Linguistic Annotation Workshop, ACL-IJCNLP '09. Association for Computational Linguistics (ACL), Stroudsburg, PA, USA, pp. 125–129.

Bhattacharyya, P., 2010. IndoWordNet, in: Proceedings of 7th Language Resources and Evaluation Conference. European Language Resources Association (ELRA), Malta, pp. 3785–3792.

Bilgin, O., Çetinoğlu, Ö., Oflazer, K., 2004. Building a Wordnet for Turkish. Romanian Journal of Information Science and Technology 7, 163–172.

Blondel, V.D., Senellart, P.P., 2002. Automatic Extraction of Synonyms in a Dictionary, in: Proceedings of the SIAM Workshop on Text Mining. Arlington, Texas, USA, pp. 1–7.

Böhmová, A., Hajič, J., Hajičová, E., Hladká, B., 2003. The Prague Dependency Treebank, in: Abeillé, A. (Ed.), Treebanks, Text, Speech and Language Technology. Springer Netherlands, pp. 103–127.

Borin, L., Forsberg, M., 2014. Swesaurus; or, the Frankenstein Approach to Wordnet Construction, in: Proceedings of the 7th Global WordNet Conference. University of Tartu Press, Tartu, Estonia, pp. 215–223.

Buscaldi, D., Rosso, P., 2008. Geo-WordNet: Automatic Georeferencing of WordNet, in: Proceedings of the 6th International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), Marrakech, Morocco, pp. 1255–1258.

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.

Chagnaa, A., Ock, C.-Y., Choe, H.-S., 2007. Extracting Features for Verifying WordNet, in: Zhang, Z., Siekmann, J. (Eds.), Knowledge Science, Engineering and Management, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 605–610.

Charoenporn, T., Sornlertlamvanich, V., Mokarat, C., Isahara, H., 2008. Semi-automatic Compilation of Asian WordNet, in: 14th Annual Meeting of the Association for Natural Language Processing. pp. 1041–1044.

Cilibrasi, R.L., Vitanyi, P.M., 2007. The Google Similarity Distance. Knowledge and Data Engineering, IEEE Transactions 19, 370–383.

Clark, A., Fox, C., Lappin, S., 2013. The handbook of computational linguistics and natural language processing. John Wiley & Sons, Singapore.

Collins, A.M., Quillian, M.R., 1969. Retrieval Time from Semantic Memory. Journal of Verbal Learning and Verbal Behaviour 8, 240–247.

Cruse, D.A., 1986. Lexical Semantics. Cambridge University Press, New York.

Davidson, S., 2013. Wordnik. The Charleston Advisor 15, 54–58.

De Lacalle, M.L., Laparra, E., Rigau, G., 2014. Predicate Matrix: Extending SemLink Through WordNet Mappings, in: Proceedings of the 9th International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), Reykjavik, Iceland, pp. 903–909.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-Scale Hierarchical Image Database, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, pp. 248–255.

De Nooy, W., Mrvar, A., Batagelj, V., 2011. Exploratory Social Network Analysis with Pajek. Cambridge University Press.

Derwojedowa, M., Piasecki, M., Szpakowicz, S., Zawisławska, M., Broda, B., 2008. Words, Concepts and Relations in the Construction of Polish WordNet, in: Proceedings of the 4th Global WordNet Conference. Szeged, Hungary, pp. 162–177.

Dyvik, H., 2004. Translations as Semantic Mirrors: From Parallel Corpus to Wordnet. Language and Computers 49, 311–326.

Esuli, A., Sebastiani, F., 2006. SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining, in: Proceedings of 5th International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), Genoa, Italy, pp. 417–422.

Farlex, I.N.C., 2009. The Free Dictionary. Retrieved June 28, 2012.

Farreres, X., Rigau, G., Rodriguez, H., 1998. Using Wordnet for Building Wordnets, in: Proceedings of COLING-ACL Workshop on Usage of WordNet in Natural Language Processing Systems. Montreal, Canada, pp. 65–72.

Fellbaum, C., 2010. WordNet, in: Poli, R., Healy, M., Kameas, A. (Eds.), Theory and Applications of Ontology: Computer Applications. Springer Netherlands, pp. 231–243.

Fellbaum, C., 2002a. On the Semantics of Troponymy, in: The Semantics of Relationships. Springer, pp. 23–34.

Fellbaum, C., 2002b. Parallel Hierarchies in the Verb Lexicon, in: Proceedings of the OntoLex Workshop. Presented at the Language Resources and Evaluation Conference (LREC), European Language Resources Association (ELRA), Las Palmas, Spain, pp. 27–31.

Fellbaum, C., 1998a. WordNet: An Electronic Lexical Database, MIT Press. ed. Wiley Online Library, Cambridge, USA.

Fellbaum, C., 1998b. A Semantic Network of English: The Mother of All Wordnets, in: EuroWordNet: A Multilingual Database with Lexical Semantic Networks. Springer, pp. 137–148.

Fellbaum, C., 1998c. A Semantic Network of English Verbs, in: WordNet: An Electronic Lexical Database. MIT Press, Cambridge, Massachusetts, USA, pp. 69–104.

Fellbaum, C. (editor), 1998. WordNet: An Electronic Lexical Database. MIT Press, Cambridge, USA.

Fellbaum, C., Vossen, P., 2008. Challenges for a Global Wordnet, in: Online Proceedings of the First International Workshop on Global Interoperability for Language Resources (ICGL 2008). Online published by Ed. Webster J., N.Ide, A.Chengyu Fang, City University of Hongkong, pp. 75–81.

Fischer, D.H., 1997. Formal Redundancy and Consistency Checking Rules for the Lexical Database WordNet 1.5, in: Workshop Proceedings of on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications. Association for Computational Linguistics (ACL), Madrid, Spain, pp. 22–31.

Fišer, D., Sagot, B., 2008. Combining Multiple Resources to Build Reliable Wordnets, in: Text, Speech and Dialogue. Springer, pp. 61–68.

Freihat, A.A., Giunchiglia, F., Dutta, B., 2013. Approaching Regular Polysemy in WordNet, in: eKNOW 2013, The Fifth International Conference on Information, Process, and Knowledge Management. pp. 63–69.

Gabrilovich, E., Markovitch, S., 2009. Wikipedia-based semantic interpretation for natural language processing. J. Artif. Intell. Res. 443–498.

Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L., 2002a. Sweetening Ontologies with DOLCE, in: Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web. Springer, pp. 166–181.

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.

Gangemi, A., Guarino, N., Oltramari, A., Borgo, S., 2002b. Cleaning-up Wordnet's Top-level, in: Proceedings of the 1st International WordNet Conference. pp. 21–25.

Gómez-Pérez, A., Benjamins, R., 1999. Overview of Knowledge, Sharing and Reuse Components: Ontologies and Problem-Solving Methods, in: Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends. CEUR Publications, Stockholm, Sweden, pp. 1–15.

Guarino, N., Welty, C., 2002. Evaluating Ontological Decisions with OntoClean. Communications of the ACM - Ontology: Different Ways of Representing the Same Concept 45, 61–65.

Gupta, P., 2002. Approaches to Checking Subsumption in GermaNet, in: Proceedings of the 3rd International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), Las Palmas, Canary Islands, Spain, pp. 8–13.

Hajic, J., Holub, M., Hucınová, M., Pavlık, M., 2004. Validating and Improving the Czech WordNet via Lexico-semantic Annotation of the Prague Dependency Treebank, in: Workshop Proceedings of the 4th International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), Lisbon, Portugal, pp. 25–30.

Havasi, C., Speer, R., Alonso, J., 2009. ConceptNet: A Lexical Resource for Common Sense Knowledge, in: Recent Advances in Natural Language Processing V. John Benjamins, Amsterdam & Philadelphia, pp. 269–280.

Hearst, M.A., 1992. Automatic Acquisition of Hyponyms from Large Text Corpora, in: Proceedings of the 14th Conference on Computational Linguistics - Volume 2, COLING '92. Association for Computational Linguistics (ACL), Stroudsburg, PA, USA, pp. 539–545.

Henrich, V., Hinrichs, E., 2010. Standardizing Wordnets in the ISO Standard LMF: Wordnet-LMF for GermaNet, in: Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics (ACL), pp. 456–464.

Hicks, A., Herold, A., 2011. Cross-lingual Evaluation of Ontologies with Rudify, in: Knowledge Discovery, Knowledge Engineering and Knowledge Management. Springer, pp. 151–163.

Hirst, G., St-Onge, D., 1998. Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms, in: WordNet: An Electronic Lexical Database. MIT Press, Cambridge, Massachusetts, USA, pp. 305–332.

Howe, D.C., 2009. RiTa.Wordnet. A software toolkit for generative literature [WWW Document]. URL http://www.rednoise.org/rita/wordnet/documentation/

Kahusk, N., Orav, H., Vare, K., 2012. Cross-linking Experience of Estonian WordNet, in: Human Language Technologies-the Baltic Perspective: Proceedings of the Fifth International Conference Baltic HLT 2012-Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 96–102.

Kahusk, N., Vider, K., 2002. Estonian WordNet Benefits from Word Sense Disambiguation, in: Proceedings of the 1t Global WordNet Conference. Mysore, India, pp. 26–31.

Kaji, H., Watanabe, M., 2006. Automatic Construction of Japanese WordNet, in: Proceedings of the 5th International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), Genoa, Italy, pp. 1262–1267.

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.

Kerner, K., Orav, H., Parm, S., 2010. Growth and revision of Estonian wordnet, in: Proceedings of 5th International Global WordNet Conference. Narosa Publisers, Mumbai, India, pp. 198–202.

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

Krovetz, R., 1997. Homonymy and Polysemy in Information Retrieval, in: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics (ACL), pp. 72–79.

Kubis, M., 2012. A Query Language for WordNet-Like Lexical Databases, in: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (Eds.), Intelligent Information and Database Systems, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 436–445.

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

Lee, C., Lee, G., Yun, S.J., 2000. Automatic WordNet Mapping Using Word Sense Disambiguation, in: Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13, EMNLP '00. Association for Computational Linguistics (ACL), Stroudsburg, PA, USA, pp. 142–147.

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.

Lin, J.-Y., Yang, C.-H., Tseng, S.-C., Huang, C.-R., 2002. The Structure of Polysemy: a Study of Multi-sense Words Based on WordNet, in: Proceedings of the 16th Pacific Asia Conference on Language, Information, and Computation. pp. 320–329.

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.

Lo, C.-S., Chen, Y.-R., Lin, C.-Y., Hsieh, S.-K., 2008. Automatic Labelling of Troponymy for Chinese Verbs, in: Proceedings of the 20th Conference on Computational Linguistics and Speech Processing. Taipei, Taiwan, pp. 284–292.

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., Tilk, O., Võhandu, L., 2013. How to Create Order in Large Closed Subsets of Wordnet-type Dictionaries. The yearbook of Estonian Association for Applied Linguistics (EAAL) 149–160.

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.
Lyons, J., 1977. Semantics (vols I & II). Cambridge. Cambridge University Press.
Mahdisoltani, F., Biega, J., Suchanek, F., 2015. YAGO3: A Knowledge Base from Multilingual WikiPedias, in: 7th Biennial Conference on Innovative Data Systems Research (CIDR). www.cidrdb.org, Asilomar, California, USA.
Martin, P., 2003. Correction and Extension of WordNet 1.7, in: Conceptual Structures for Knowledge Creation and Communication. Springer, pp. 160–173.
Maziarz, M., Piasecki, M., Szpakowicz, S., 2012. Approaching plWordNet 2.0, in: Proceedings of the 6th Global Wordnet Conference, Matsue, Japan. pp. 189–196.
Měchura, M.B., 2010. What WordNet Does Not Know about Selectional Preferences? Proceedings of the 14th Euralex International Congress 431–436.
Mihalcea, R., 2003. Turning Wordnet into an Information Retrieval Resource: Systematic Polysemy and Conversion to Hierarchical Codes. International Journal of Pattern Recognition and Artificial Intelligence 17, 689–704.
Mihalcea, R., Moldovan, D.I., 2001. EZ. WordNet: Principles for Automatic Generation of a Coarse Grained WordNet, in: FLAIRS Conference. pp. 454–458.
Miller, G.A., 1998. Nouns in WordNet, in: WordNet: An Electronic Lexical Database. MIT Press, Cambridge, Massachusetts, USA, pp. 24–45.
Miller, G.A., Fellbaum, C., 2007. WordNet Then and Now. Language Resources and Evaluation 41, 209–214.
Miller, T., Gurevych, I., 2014. WordNet–Wikipedia–Wiktionary: Construction of a Three-way Alignment, in: Proceedings of the 9th International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), Reykjavik, Iceland, pp. 2094–2100.
Mititelu, V.B., 2006. Automatic Extraction of Patterns Displaying Hyponym-hypernym Co-occurrence from Corpora. Proceedings of the First Central European Student Conference in Linguistics.
Montazery, M., Faili, H., 2010. Automatic Persian WordNet Construction, in: Proceedings of the 23rd International Conference on Computational Linguistics: Posters. Association for Computational Linguistics (ACL), pp. 846–850.
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.

Navigli, R., Ponzetto, S.P., 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artif. Intell. 193, 217–250.

Navigli, R., Ponzetto, S.P., 2010. BabelNet: Building a Very Large Multilingual Semantic Network, in: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10. Association for Computational Linguistics (ACL), Stroudsburg, PA, USA, pp. 216–225.

Nikulásdóttir, A.B., Whelpton, M., 2009. Automatic Extraction of Semantic Relations for Less-Resourced Languages, in: Proceedings of the NODALIDA 2009 Workshop: Wordnets and Other Lexical Semantic Resources between Lexical Semantics, Lexicography, Terminology and Formal Ontologies. Odense, Denmark, pp. 1–6.

Niles, I., Pease, A., 2003. Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology, in: Proceedings of the IEEE International Conference on Information and Knowledge Engineering. pp. 412–416.

Noy, N., 2003. The OntoClean Ontology in Protégé.

Oakes, M.P., 2005. Using Hearst's Rules for the Automatic Acquisition of Hyponyms for Mining a Pharmaceutical Corpus, in: RANLP Text Mining Workshop. pp. 63–67.

Oltramari, A., Gangemi, A., Guarino, N., Masolo, C., 2002. Restructuring WordNet's Top-level: The OntoClean Approach, in: Workshop Proceedings of OntoLex. Presented at the Proceedings of 3rd Language Resources and Evaluation Conference, European Language Resources Association (ELRA), pp. 17–26.

Pala, K., Smrž, P., 2004. Building Czech Wordnet. Romanian Journal of Information Science and Technology 7, 79–88.

Panchenko, A., Morozova, O., Naets, H., 2012. A Semantic Similarity Measure Based on Lexico-syntactic Patterns, in: Proceedings of the 11th Conference on Natural Language Processing. pp. 174–178.

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.

Pedersen, T., Patwardhan, S., Michelizzi, J., 2004. WordNet::Similarity: Measuring the Relatedness of Concepts, in: Demonstration Papers at HLT-NAACL 2004, HLT-NAACL–Demonstrations '04. Association for Computational Linguistics (ACL), Stroudsburg, PA, USA, pp. 38–41.

Peters, W., Guthrie, L., Wilks, Y., 2002. Cross-Linguistic Discovery of Semantic Regularity, in: Proceedings of the 1st Global WordNet Conference. Mysore, India, pp. 360–368.

Peters, W., Peters, I., Vossen, P., 1998. Automatic Sense Clustering in EuroWordNet, in: Proceedings of the 1st International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), Granada, Spain, pp. 409–416.

Pethö, G., 2001. What is Polysemy? A Survey of Current Research and Results. Pragmatics and the Flexibility of Word Meaning 175–224.

Piasecki, M., Burdka, L., Maziarz, M., 2013a. Wordnet Diagnostics in Development, in: Human Language Technologies as a Challenge for Computer Science and Linguistics. Poznań, Poland, pp. 268–272.

Piasecki, M., Marcińczuk, M., Ramocki, R., Maziarz, M., 2013b. WordNetLoom: a WordNet Development System Integrating Form-based and Graph-based Perspectives. Int. J. Data Min. Model. Manag. 5, 210–232.

Piasecki, M., Szpakowicz, S., Broda, B., 2009. A Wordnet from the Ground Up. Oficyna Wydawnicza Politechniki Wrocławskiej.

Piasecki, M., Szpakowicz, S., Fellbaum, C., Pedersen, B.S., 2013c. Introduction to the special issue: On wordnets and relations. Lang. Resour. Eval. 47, 757.

Prabhu, V., Desai, S., Redkar, H., Prabhugaonkar, N., Nagvenkar, A., Karmali, R., 2012. An Efficient Database Design for IndoWordNet Development Using Hybrid Approach, in: Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing. Association for Computational Linguistics (ACL), Mumbai, India, pp. 229–235.

Prószéky, G., Miháltz, M., 2002. Semi-automatic Development of the Hungarian WordNet, in: Proceedings of the 3rd International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), Las Palmas, Canary Islands, Spain.

Pustejovsky, J., 1991. The Generative Lexicon. Computational Linguistics 17, 409–441.

Ramachandran, D., Reagan, P., Goolsbey, K., 2005. First-Orderized ResearchCyc: Expressivity and Efficiency in a Common-Sense Ontology, in: AAAI Workshop on Contexts and Ontologies: Theory, Practice and Applications. The AAAI Press, Menlo Park, California, pp. 33–40.

Resnik, P., Lin, J., 2010. Evaluation of NLP Systems, in: The Handbook of Computational Linguistics and Natural Language Processing. John Wiley & Sons, Singapore, pp. 271–295.

Reynaud, C., Safar, B., 2007. Exploiting WordNet as Background Knowledge, in: Proceedings of the Workshop on Ontology Matching at ISWC/ASWC2007. Presented at the 2nd International Workshop on Ontology Matching, Busan, South Korea, pp. 291–295.

Richens, T., 2011. Lexical Database Enrichment through Semi-Automated Morphological Analysis. Aston University, UK.

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.

Rizov, B., 2008. Hydra: a Modal Logic Tool for Wordnet Development, Validation and Exploration, in: Proceedings of the 6th International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), Marrakech, Morocco, pp. 1523–1528.

Rodríguez, H., Climent, S., Vossen, P., Bloksma, L., Peters, W., Alonge, A., Bertagna, F., Roventini, A., 1998. The Top-Down Strategy for Building EuroWordNet: Vocabulary Coverage, Base Concepts and Top Ontology. Computers and the Humanities 32, 117–152.

Ruiz-Casado, M., Alfonseca, E., Castells, P., 2005a. Automatic Extraction of Semantic Relationships for Wordnet by Means of Pattern Learning from WikiPedia, in: Natural Language Processing and Information Systems. Springer, pp. 67–79.

Ruiz-Casado, M., Alfonseca, E., Castells, P., 2005b. Automatic Assignment of Wikipedia Encyclopedic Entries to WordNet Synsets, in: Szczepaniak, P.S., Kacprzyk, J., Niewiadomski, A. (Eds.), Advances in Web Intelligence, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 380–386.

Rydin, S., 2002. Building a Hyponymy Lexicon with Hierarchical Structure, in: Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition-Volume 9. Association for Computational Linguistics (ACL), pp. 26–33.

Sagot, B., Fišer, D., 2012. Cleaning Noisy Wordnets, in: Proceedings of the 8th International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), Istanbul, Turkey, pp. 23–25.

Sagot, B., Fišer, D., 2011. Extending Wordnets by Learning from Multiple Resources, in: LTC'11: 5th Language and Technology Conference. Poznan, Poland.

Sagri, M.T., Tiscornia, D., Bertagna, F., 2004. Jur-WordNet, in: Proceedings of the 2nd International Global Wordnet Conference. pp. 305–310.

Saito, J.-T., Wagner, J., Katz, G., Reuter, P., Burke, M., Reinhard, S., 2002. Evaluation of GermaNet: Problem Using GermaNet for Automatic Word Sense Disambiguation, in: Proceedings of the LREC Workshop on WordNet Structure and Standardization and How These Affect WordNet Applications and Evaluation. European Language Resources Association (ELRA), Las Palmas, Canary Islands, Spain, pp. 14–29.

Sang, E.T.K., 2007. Extracting Hypernym Pairs from the Web, in: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions. Association for Computational Linguistics (ACL), pp. 165–168.

Saveski, M., Trajkovski, I., 2010. Automatic Construction of Wordnets by Using Machine Translation and Language Modelling, in: 13th Multiconference Information Society. Ljubljana, Slovenia.

Sinopalnikova, A., 2004. Word Association Thesaurus as a Resource for Building Wordnet, in: Proceedings of the 2nd International WordNet Conference. pp. 199–205.

Smith, B., Fellbaum, C., 2004. Medical WordNet: A New Methodology for the Construction and Validation of Information Resources for Consumer Health, in: Proceedings of the 20th International Conference on Computational Linguistics, COLING '04. Association for Computational Linguistics (ACL), Stroudsburg, PA, USA.

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

Snow, R., Jurafsky, D., Ng, A.Y., 2004. Learning Syntactic Patterns for Automatic Hypernym Discovery, in: Advances in Neural Information Processing Systems 17. MIT Press, Cambridge, Massachusetts, USA, pp. 1297–1304.

Suchanek, F.M., Kasneci, G., Weikum, G., 2008. YAGO: A Large Ontology from Wikipedia and WordNet. Web Semantics: Science, Services and Agents on the World Wide Web, World Wide Web Conference 2007Semantic Web Track 6, 203–217.

Suchanek, F.M., Kasneci, G., Weikum, G., 2007. YAGO: A Core of Semantic Knowledge Unifying Wordnet and Wikipedia, in: 16th International World Wide Web Conference, WWW. pp. 697–706.

Sure, Y., Angele, J., Staab, S., 2003. OntoEdit: Multifaceted Inferencing for Ontology Engineering, in: Spaccapietra, S., March, S., Aberer, K. (Eds.), Journal on Data Semantics I, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 128–152.

Tengi, R.I., 1998. Design and Implementation of the WordNet Lexical Database and Searching Software, in: WordNet: An Electronic Lexical Database. MIT Press, Cambridge, Massachusetts, USA, pp. 105–127.

Thoongsup, S., Robkop, K., Mokarat, C., Sinthurahat, T., Charoenporn, T., Sornlertlamvanich, V., Isahara, H., 2009. Thai WordNet Construction, in: Proceedings of the 7th Workshop on Asian Language Resources. Association for Computational Linguistics (ACL), pp. 139–144.

Tufis, D., Cristea, D., Stamou, S., 2004a. BalkaNet: Aims, Methods, Results and Perspectives. A General Overview. Romanian Journal of Information Science and Technology 7, 9–43.

Tufis, D., Ion, R., Barbu, E., Barbu, V., 2004b. Cross-lingual Validation of Multilingual Wordnets, in: Proceedings of the 2nd Global WordNet Conference. pp. 332–340.

Vercruysse, S., Kuiper, M., 2013. WordVis: JavaScript and Animation to Visualize the WordNet Relational Dictionary, in: Kudělka, M., Pokorný, J., Snášel, V., Abraham, A. (Eds.), Proceedings of the Third International Conference on Intelligent Human Computer Interaction (IHCI 2011), Prague, Czech

Republic, August, 2011, Advances in Intelligent Systems and Computing. Springer Berlin Heidelberg, pp. 137–145.
Verdezoto, N., Vieu, L., 2011. Towards Semi-automatic Methods for Improving WordNet, in: Proceedings of the Ninth International Conference on Computational Semantics. Association for Computational Linguistics (ACL), pp. 275–284.
Vetulani, Z., Kubis, M., Obrêbski, T., 2010. PolNet-Polish WordNet: Data and Tools, in: Proceedings of the 7th International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA), Malta, pp. 3793–3797.
Vider, K., 2001. Eesti keele tesaurus-teooria ja tegelikkus, in: Ettekannete Kogumik. Presented at the Leksikograafiaseminar“ Sõna tänapäeva maailmas”/Leksikografinen seminaari“ Sanat nykymaailmassa,” Eesti Keele Instituut, Tallinn, Eesti, pp. 134–156.
Völker, J., Vrandečić, D., Sure, Y., 2005. Automatic Evaluation of Ontologies (AEON), in: The Semantic Web–ISWC 2005. Springer, pp. 716–731.
Vossen, P., 2004. EuroWordNet: A Multilingual Database of Autonomous and Language-Specific Wordnets Connected via an Inter-Lingual Index. International Journal of Lexicography 17, 161–173.
Vossen, P., 1998a. Introduction to EuroWordNet, in: EuroWordNet: A Multilingual Database with Lexical Semantic Networks. Springer, pp. 1–17.
Vossen, P., 1998b. Introduction to EuroWordNet. Computers and the Humanities 32, 73–89.
Vossen, P., Bloksma, L., Rodriguez, H., Climent, S., Calzolari, N., Roventini, A., Bertagna, F., Alonge, A., Peters, W., 1998. The EuroWordNet Base Concepts and Top Ontology. Centre National de la Recherche Scientifique, Paris, France.
Wierzbicka, A., 1984. Apples Are Not a “kind of fruit”: The Semantics of Human Categorization. American Ethnologist 11, 313–328.

ABSTRACT

The goal of this thesis is to prove that semantic hierarchies of wordnet-type dictionaries do contain yet undiscovered substructures which correspond to certain descriptions (test patterns). The usage of these patterns to validate semantic hierarchies may improve wordnet structure significantly.

More precisely, this thesis studies test patterns that contain the *multiple inheritance* cases (i.e. cases where one synset has many parents and that correspond to polysemy in the lexical semantics). Here, test patterns are examined and every case of applying a test pattern to the wordnet hierarchy is termed as a test pattern instance. *Multiple inheritance* plays an important role because it is often prone to different semantic errors.

Every test pattern represents different perspective to the substructures of the semantic hierarchy, pointing to different type of possible errors in it. All test patterns together cover all multiple inheritance cases in a semantic hierarchy several times.

The main research method in this dissertation is pattern-based validation. The whole process is conducted as follows: programs created by the author of this thesis will detect each instance of the test patterns and these are automatically stored in a file; the lexicographer validates each case and makes corrections, if necessary, using the wordnet management system.

One of the most important results of this thesis are descriptions of **mathematical models or graphs that represent yet undiscovered substructures (test patterns) of the semantic hierarchies of the wordnet-type dictionaries**.

This work associates every test pattern with possible types of errors and equip them with usage examples.

All test patterns are cross-language, i.e. they are applicable to every language wordnet (approximately 50 different languages).

Second important result is **test patterns as the tool of lexicographer** in validation process. Test patterns simplify the inspecting of the semantic hierarchies of wordnet significantly, because they point to substructures that may need the correction by lexicographers.

Third important result **is an overview of 10 versions of Estonian Wordnet iterative evolution** where test patterns were employed and which prove the efficiency of test patterns in validation of the semantic hierarchies of wordnet. For instance, from version 66 to version 70, the number of Estonian Wordnet multiple inheritance cases (it means also the number of test patterns instances) in the semantic hierarchies of IS-A relation are decreased approximately 97%.

KOKKUVÕTE

Doktoritöö **eesmärk** on tõestada, et *wordnet*-tüüpi suursõnastike semantilistes hierarhiates esineb seni avastamata alamstruktuure, mida on mõistlik kasutada hierarhiate kontrollimiseks ja valideerimiseks ning mille järjepideval – iga uue *wordnet*'i versiooniga – rakendamisel paraneb suuresti semantiliste hierarhiate struktuur.

Täpsemalt uuritakse semantiliste hierarhiate alamstruktuure, mis sisaldavad mitmese pärimise juhtumeid, st selliseid, kus ühel tipul on mitu vanemat ja millele leksikaalses semantikas vastab polüseemia. Töö kontekstis nimetatakse niisuguseid kindla kujuga graafi alamstruktuure testmustriteks (vaadeldavad kui klassid objektorienteeritud lähenemises). Igat konkreetset juhtu, mis on testmustri kaudu *wordnet*'i hierarhilisest struktuurist eraldatud, nimetatakse isendiks. Mitmene pärimine on aga siinjuures oluline sellega seotud võimalike semantiliste vigade tõttu.

Iga testmuster esindab erisugust vaadet semantilise hierarhia alamstruktuuridele, osutades neis võimalikele eri tüüpi vigadele. Kõikide testmustrite korraga kasutamisel kaetakse semantilise hierarhia kõik mitmese pärimise juhud mitmekordselt.

Uurimismeetodina kasutakse mustritepõhist hindamist. Kogu protsess toimub järgmiselt: töö autori loodud programmide abil leitakse igale testmustrile isendid, mis salvestatakse automaatselt faili; leksikograaf hindab igat juhtumit ja teeb vajaduse korral korrektuurid *wordnet*'i semantilises hierarhias selle haldamissüsteemi abil.

Töö **peamine tulemus** on *wordnet*-tüüpi sõnastike semantilises hierarhias leiduvate ja seni avastamata **alamstruktuuride (testmustrite) kirjeldused matemaatiliste mudelite ehk graafidena**.

Töös seostatakse iga testmuster võimalike veatüüpidega, mida need on suutelised avastama ja esitatakse kasutusnäited.

Kõik **testmustrid on keelte ülesed**, st rakendatavad iga keele (*ca* 50) *wordnet*'i puhul.

Töö teine väga tähtis **tulemus** on **testmustrid kui leksikograafi tööriist** *wordnet*'i semantilise hierarhia valideerimisel. Testmustrid lihtsustavad tuntavalt *wordnet*'i hierarhiate läbivaatamist, sest neile on omane osutada vaid hierarhia sellistele piirkondadele, kus asub võimalik leksikograafi korrigeerivat sekkumist vajav koht.

Töö kolmas oluline tulemus on **ülevaade Eesti Wordneti kümne versiooni testmustrite rakendamise arengust**, mis kinnitab testmustrite efektiivsust *wordnet*'i semantiliste hierarhiate korrigeerimisel. Näiteks, alates Eesti Wordneti 66. versioonist kuni 70. versioonini on mitmese pärimise juhtude arv (ka testmustrite isendite arv) langenud ≈ 97%.

ACKNOWLEDGMENTS

To begin with, I would like to extend a heartfelt thank you to the friendly and wonderful collective at the TUT department of informatics. I am grateful to Jüri Vilipõld, who noticed me in an informatics module for civil engineering students and invited me to work at the department of informatics. I am also obliged to my supervisor Leo Võhandu, who injected me with passion for working in science.

I would also like to thank the University of Tartu Wordnet workgroup for providing positive feedback on the methods employed in this thesis. Thank you, Kadri, Heili, Neeme and Kadri.

I wish to extend gratitude to Margit Langemets with whom I was able to discuss systematic polysemy in the context of this thesis and who also read and gave her evaluation of the theoretical part of this work.

I am also grateful to all the language editors who much improved the text of my thesis. Thank you, Ailin, Meeli and Owain.

I would like to give special thanks to my parents Rein and Õie who taught me to work hard since childhood.

Finally, I am most thankful to my dear wife Aive, who was consistently willing to forego her plans for the sake of my work. I am also grateful for my lovely children Andri Ruuben and Hanna Raahel, whose joys and woes alleviated the monotony of the work process. Your part in this work, my dear family, is invaluable!

APPENDIX A

Lohk, A.; Vare, K.; Võhandu, L. (2012). First Steps in Checking and Comparing Princeton WordNet and Estonian Wordnet. In: Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH: EACL 2012; April 23 - 24 2012; Avignon France. 2012, pp. 25 - 29.

First steps in checking and comparing Princeton WordNet and Estonian Wordnet

Ahti Lohk
Tallinn University of Technology
Raja 15-117
Tallinn, ESTONIA
[ahti.lohk@ttu.ee]

Kadri Vare
University of Tartu
Liivi 2-308
Tartu, ESTONIA
[kadri.vare@ut.ee]

Leo Võhandu
Tallinn University of Technology
Raja 15-117
Tallinn, ESTONIA
[leov@staff.ttu.ee]

Abstract

Each expanding and developing system requires some feedback to evaluate the normal trends of the system and also the unsystematic steps. In this paper two lexical-semantic databases – Princeton WordNet (PrWN) and Estonian Wordnet (EstWN)- are being examined from the visualization point of view. The visualization method is described and the aim is to find and to point to possible problems of synsets and their semantic relations.

1 Introduction

Wordnets for different languages have been created for a quite a long time [1]; also these wordnets have been developed further and updated with new information. Typically there is a special software for editing wordnets, for example VisDic[2], WordnetLoom (Piasecki et al 2010), Polaris (Louw, 1998). These editing tools often present only one kind of view of the data which might not be enough for feedback or for detecting problematic synsets/semantic relations. The visualization method described here can be used separately from the editing tool; therefore it provides an additional view to data present in wordnet.

For initial data PrWN version 3.0[3] and EstWN version 63[4] have been taken. PRWN contains of 117 374 synsets and EstWn of 51 688 synsets. The creation of EstWN started in 1998 within the EuroWordNet project[5]. At present the main goal is to increase EstWN with new concepts and enrich EstWN with different kinds of semantic relations. But at the same time it is necessary to check and correct the concepts already present (Kerner, 2010).

The main idea and basic design of all wordnets in the project came from Princeton WordNet (more in Miller et al 1990). Each wordnet is structured along the same lines: synonyms (sharing the same meaning) are grouped into synonym sets (synsets). Synsets are connected to each other by semantic relations, like hyperonymy (is-a) and meronymy (is-part-of). As objects of analysis only noun synsets and hyperonymy-hyponymy relations are considered (of course, it is possible to extend the analysis over different word classes and different semantic relations). So, due to these constraints we have taken 82 115 synsets from PRWN (149 309 different words in synsets) and 41 938 synsets from EstWN (64 747 different words in synsets).

2 Method

We will explain our method's main idea with a small artificial example. Let us have a small separated subset presented as a matrix:

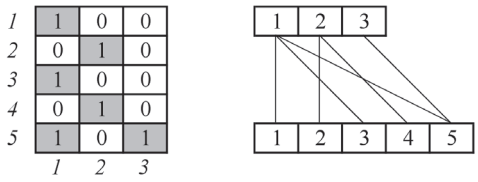


Figure 1. Relation-matrix and bipartite graph

In the rows of that table we have synsets and in columns hyperonyms. On the right side of

[1] http://www.globalwordnet.org/
[2] http://deb.fi.muni.cz/clients-debvisdic.php
[3] http://wordnet.princeton.edu/
[4] http://www.cl.ut.ee/ressursid/teksaurus/
[5] http://www.illc.uva.nl/EuroWordNet/

that figure we have presented the same data as a bipartite graph where all column numbers are positioned on the upper line and all rows on the lower line. Every connecting line on the right side has been drawn between every "1"-s column and row number. As we see a lot of line crossings there exist even in our very small example. It is possible to reorder the rows and columns of that table into optimal positions so that the number of line crossings would be minimal possible. If there is full order then there will be no crossings of lines.

Generally this crossing number minimization is a NP-complete task. We are using the idea of Stephan Niermann's (2005) evolutionary algorithm to minimize the number of line crossings.

In our example the optimal result will be:

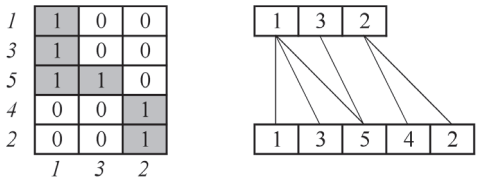


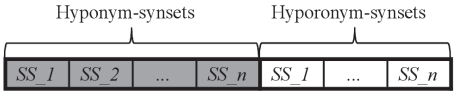
Figure 2. Reordered (arranged) relation-matrix and bipartite graph

As we can see there are no crossings and all connections are separated into two classes – let's call them closed sets. We have got a nice and natural ordering for rows and columns. With that kind of picture the relations between words (synsets) are easier to see and understand. We will present real cases from PrWN and EstWN later.

3 Practical application of the method

Next we will describe the steps that should be taken in order to obtain visual pictures for lexicographers.

- First the word class and a semantic relation of interest is chosen from wordnet. For nouns and verbs hyperonymy and hyponymy are probably the most informative relations, for adjectives and adverbs near_synonymy (but of course this method allows us to choose different semantic relations in combination with different word classes).
- In order to find closed sets we use the connected component separating algorithm for graphs given in D. Knuth (1968). For example using hyponym-hyperonym relation and word classes of nouns then there will be 7 907 closed sets for EstWN and 15 452 closed sets for PrWN. Every closed set is presented in a table as a row with different lengths. An arbitrary closed set is similar to the following picture in Figure 3.



SS1 - synset 1, SS2 - synset 2, ...

Figure 3. Example of a closed set

- As a next step we use all connections for those two sets in a wordnet to get the relation matrix as it is shown in Figure 1 left part.
- Then the minimal crossing algorithm is used (result is seen on the right side of Figure 2).
- As the last step a lexicographer analyzes the figures.

It is still important to mention that our approach is not quite useful for analyzing the large closed sets. The reason is that in Nierman's evolutionary algorithm if the size of the matrix grows than the time increases with the speed $O(n^2)$. For example, to solve the 30x30 matrix, it takes 3 minutes and to solve 60x60 matrix, it takes 60 minutes. That is the reason why in this paper only closed sets that do not exceed the 30 hyponym sets are considered. The pictures from closed sets (Figure 4, 5, 6) were solved as follows: Figure 4 (3 x 5 matrix) 0,28sec, Figure 5 (4 x 11 matrix) 1,5sec, Figure 6 (4 x 12 matrix) 1,7sec.

For larger closed sets it is better to use the modified Power Iteration Clustering method by Lin and Cohen (2010) instead of Niermann's algorithm.

As a matter of fact, the largest closed set in EstWN has 4103 hyponyms-synsets x 405 hyperonym-synsets and the largest closed set in PrWN has 2371 hyponyms-synsets x 167 hyperonym-synsets (Figure 3). As for large closed sets, it could be sensible to use only the relation matrix (Figure 2, left side) to detect where possible problematic places occur.

4 Intermediate results

In this paper we focus on the synsets having two or more hyperonyms, which is the reason of closed sets, since it is more likely to find problematic places in these synsets.

For example in EstWN only one hyperonym for a synset should ideally exist (Vider, 2001). In EstWN there are currently 1 674 concepts with two hyperonyms, 145 concepts with three or more hyperonyms and the concept which has the most hyperonyms - 9 - is ’alkydcolour’.

In PrWN there are 1 442 concepts with two hyperonyms, 34 concepts with three or more hyperonyms and the concept with the most hyperonyms – 5 – is 'atropine'.

Of course in wordnets a synset can have multiple hyperonyms in many cases, in EstWN many of the onomatopoetic words, for example (typically they have hyperonyms which denote movement and sound). But also there are cases where one of the hyperonyms is in some ways more suitable than another. Even if a synset has multiple hyperonyms a cluster still often presents a homogeneous semantic field.

One of the purposes of the visual pictures is to help in detecting so called human errors, for example:

- in a situation where in the lexicographic (manual) work a new and more precise hyperonym is added during editing process but the old one is not deleted;
- lexicographer could not decide which hyperonym fits better;
- lexicographer has connected completely wrong senses (or words) with hyperonymy relation;
- lexicographer has not properly completed the domain-specific synsets etc.

The first three points can indicate the reason of why one synset has multiple hyperonym-synsets.

For example, in Figure 4 all the members of the cluster seem to form a typical set of allergic and hypersensitivity conditions and illnesses. In EstWN currently allergies and diseases caused by allergies do not form such a cluster, because they do not share hyperonyms. But also different clusters exist where some problems can appear.

For example, in Figure 5 where all the other characters (suicide bomber, terrorist, spy etc) except ‘programmer’ are bad or criminal by their nature. This leads to a thought that maybe ‘programmer’ as a hyperonym to ‘hacker’ and ‘cracker’ is not the best; it might be that ‘programmer’ is connected with some other semantic relation.

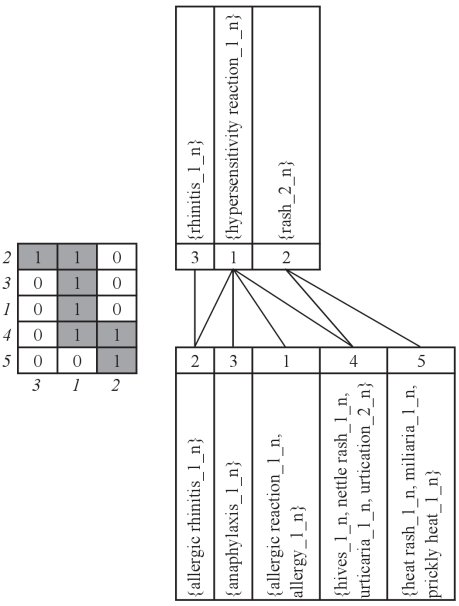


Figure 4. Rearranged bipartite graph, PrWN

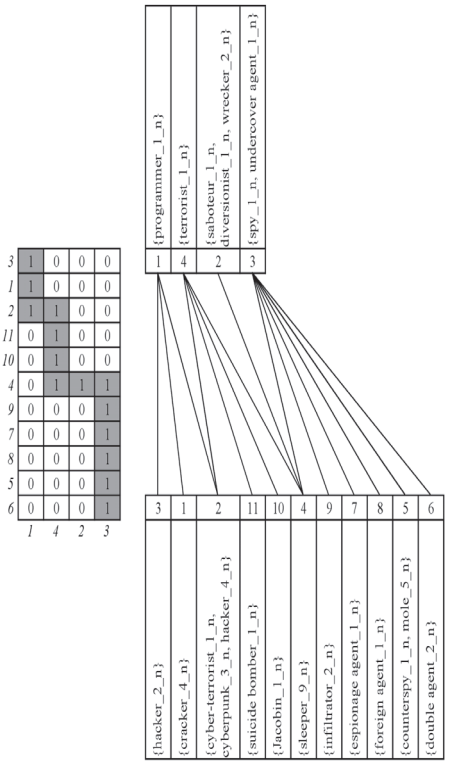


Figure 5. Rearranged bipartite graph, PrWN

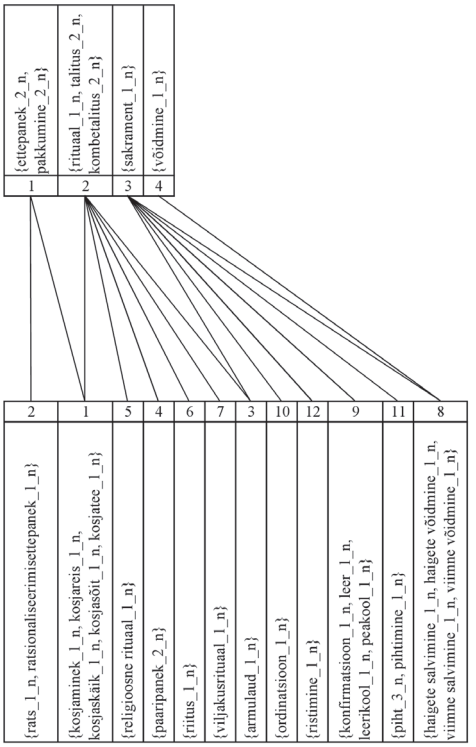


Figure 6. Rearranged bipartite graph, EstWN

Hyperonym-synsets:
1. ettepanek, pakkumine - proposal
2. rituaal, talitus, ... - rituaal
3. sakrament - sacrament
4. võidmine - unction, anoiting

Hyponym-synsets:
4. paaripanek - marriage ritual
6. riitus - rite
7. viljakusrituaal - fertility rite
3. armulaud - Holy Communion
10. ordinatsioon - ordination
12. ristimine - baptism
9. konformatsioon, ... - confirmation
11. piht, pihtimine - confession
8. haigete salvimine, ... - extreme unction
2. rats, ratsionaliseerimisettepanek - proposal for rationalization
1. kosjaminek, kosjareis, ... - a visit to bride's house to make a marriage proposal
5. religioosne rituaal - religious ritual

From EstWN many problematic synsets and/or semantic relations were discovered by using this method. In Figure 6, for example, from EstWN there is an example of a closed set for nouns. It can be seen that the word *ratsionaliseerimisettepanek* ('proposal to rationalization') does not belong to this semantic field (this semantic field can be named 'different kinds of rituals' for example). It is strange that words *ratsionaliseerimisettepanek* ('proposal to rationalization') and *kosjakäik* ('a visit to bride's house to make a marriage proposal') belong to the same closed set. Both these synsets share a hyperonym *ettepanek* ('proposal'), but *kosjakäik* should be connected to *ettepanek* ('proposal') by is_involved relation and the hyperonym to *kosjakäik* should be 'ritual' instead.

Also the relation of hyperonyms *võidmine* ('unction') and *sakrament* ('sacrament'). should be interesting. It can be seen that all the semantic relations of hyperonym *võidmine* ('unction') belong actually to *sakrament* ('sacrament'). So it is possible to state that sacrament should be hyperonym to unction. Another question arises with the word *armulaud* ('Holy Communion'). In principle, this word is correctly connected to both sacrament and ritual, but still – all of the hyponyms of sacrament are some sorts of services. These connections are probably missing from the system.

In addition, a minor detail – although *abielu* ('marriage') belongs to sacrament, it is in EstWN categorized only as a ritual and not even directly but implicitly by the word *paaripanek* ('marriage ritual')

5 Conclusion

In order to find mistakes from closed sets it is not necessary to use a bipartite graph. In some cases only the relation-matrix will be enough (Figure 1,2 left side). Clear created groupings can be considered as an advantage of bipartite graphs, which present the hyponym synsets connecting the hyperonym synsets. Often these connections can turn out as the problematic ones. Sometimes it is necessary to use the wordnet database in order to move a level up to understand the meaning of a synset.

Out of the 20 arbitrarily extracted closed sets 6 seemed to have some problems. And in PrWN there were 185 closed sets with hyperonym synsets having at least three hyperonyms. This seems to be a promising start towards using visual pictures. The situation is similar in EstWN, and since EstWN is far from "being completed" then this method has already

proven useful for lexicographers in the revision work.

To conclude, the structured bipartite figures are informative in following ways:

- It is possible to use different kinds of semantic relations to create closed sets.
- It is possible to detect subgroups.
- It is possible to detect wrong and missing semantic relations.

Acknowledgments

In this paper Kadri Vare is supported by META-NORD project (CIP-ICT-PSP.2010-4 Theme 6: Multilingual Web: Machine translation for the multilingual web); Estonian Ministry of Education and Research (Target financed research theme SF0180078s08, "Development and implementation of formalisms and efficient algorithms of natural language processing for the Estonian language") and National Programme for Estonian Language Technology.

References

Ashok K. Chandra, Dexter C. Kozen, and Larry J.Stockmeyer. 1981. Alternation. Journal of the Association for Computing Machinery, 28(1):114-133.

Association for Computing Machinery. 1983. Computing Reviews, 24(11):503-512.

Dan Gusfield. 1997. Algorithms on Strings, Trees and Sequences. Cambridge University Press, Cambridge, UK.

Donald E. Knuth. 1968, *Fundamental Algorithms, vol. 1 of Art of Computer Programming* (Reading, MA, Addison-Wesley), §2.3.3.

Frank Lin and William W. Cohen. 2010. *Power Iteration Clustering* in ICML-2010.

George Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross and Kathrine Miller. 1990. *Introduction to WordNet: An On-line Lexical database.* – International Journal of Lexicography 3, 235-312.

Kadri Kerner, Heili Orav and Sirli Parm. 2010. Growth and Revision of Estonian WordNet. *In: Principles, Construction and Application of Multilingual Wordnets.* Proceeding of the 5th Global Wordnet Conference: 5th Global Wordnet Conference; Mumbai, India. (Ed.) Bhattacha-ryya, P.; Fellbaum, Ch.; Vossen, P. Mumbai, India: Narosa Publishing House, pp 198-202.

Kadri Vider. 2001. Eesti keele tesaurus - teooria ja tegelikkus Leksikograafiaseminar *"Sõna tänapäeva maailmas" Leksikografinen seminaari "Sanat nykymaailmassa". Ettekannete kogumik*. Toim. M. Langemets. Eesti Keele Instituudi toimetised 9. Tallinn, lk 134-156.

Michael Louw. 1998. *Polaris User's Guide*. Technical report, Lernout & Hauspie . Antwerp, Belgium.

Maciej Piasecki, Michal Marcinczuk, Adam Musial, Radoslav Ramocki and Marek Maziarz. 2010. *WordnetLoom: a Graph-based Visual Wordnet Development Framework.* In Proceedings of IMCSIT, 469-476.

Stefan Niermann. 2005. Optimizing the Ordering of Tables With Evolutionary Computation. *The American Statistician*, 59(1):41-46.

APPENDIX B

Lohk, A.; Tilk, O.; Võhandu, L. (2013). How to create order in large closed subsets of wordnet-type dictionaries. Eesti Rakenduslingvistika Ühingu aastaraamat, 9, pp. 149 - 160.

doi:10.5128/ERYa9.10

EESTI RAKENDUSLINGVISTIKA ÜHINGU AASTARAAMAT 9, 149–160

HOW TO CREATE ORDER IN LARGE CLOSED SUBSETS OF WORDNET-TYPE DICTIONARIES

Ahti Lohk, Ottokar Tilk, Leo Võhandu

Abstract. This article presents a new two-step method to handle and study large closed subsets of WordNet-type dictionaries with the goal of finding possible structural inconsistencies. The notion of closed subset is explained using a WordNet tree. A novel and very fast method to order large relational systems is described and compared with some other fast methods. All the presented methods have been tested using Estonian[1] and Princeton WordNet[2] largest closed sets.

Keywords: thesaurus, closed set, seriation, Power Iteration Clustering (PIC), reducing number of crossings, WordNet

1. Introduction

There are more than 60 WordNets in the world[3]. The main idea and basic design of all these lexical resources came from Princeton WordNet (more in Miller et al. 1990). Each WordNet is structured along the same lines: synonyms (sharing the same meaning) are grouped into synonym sets (synsets). Synsets are connected to each other by semantic relations, like hyperonymy (IS-A) and meronymy (IS-PART-OF). In this article only hyperonymy-hyponymy relations are considered as objects of analysis. Of course, it is easy to extend the analysis over different word classes and different semantic relations.

WordNet has been used for a number of different purposes in information systems, including word sense disambiguation (Li et al. 1995), information retrieval (Rila et al. 1998), automatic text classification and structuring (Morato et al. 2004), automatic text summarization, natural language generation (Jing et al. 1998), machine translation (Khan et al. 2009) and even language teaching applications (Morato et al. 2004). A description of the Estonian WordNet and its properties has been given by Orav et al. (2011).

In applications where WordNet usage is considerable, the quality of the result depends on the quality of the WordNet used. Our analysis shows clearly that many

[1] Estonian WordNet: http://www.cl.ut.ee/ressursid/teksaurus/test/estwn.cgi.et (08.01.2013).
[2] Princeton WordNet: http://wordnet.princeton.edu/ (08.01.2013).
[3] The Global WordNet Association: http://www.globalwordnet.org/gwa/wordnet_table.html (08.01.2013).

WordNet-type dictionaries have a large closed subset (Table 1) caused by such semantic relations where one synset has connections to more than one supersynset. Liu et al. analyse mistakes in WordNet structures that arise particularly in cases where a synset has more than one sypersynsets (Liu et al. 2004). Richens extends the ideas of Liu et al. and presents a list of anomalies in the WordNet verb hierarchy and methods for finding them (Richens 2008)[4]. Vider (2001) proposes that in the best case every synset has only one supersynset. Closed subsets with more than one supersynset refer to possible causes of errors (Richens 2008, Lohk et al. 2012a). We present a convenient tool to study the possible structural inconsistencies of such large separated subsets.

For every synset in WordNet we have a matrix representation, as in Figure 3, upper level on the right. As we have to deal with very big matrices one needs a well ordered final representation of such a matrix to understand its hidden structure. Our goal is to reorder that matrix into the form of Figure 3, lower level on the right. That representation corresponds to the so-called Multidimensional Scale representation in psycholinguistics.

In the next section we explain the content of a closed set (Lohk et al. 2012b).

2. Closed sets

The synsets of WordNet-type dictionaries have as semantic connections hierarchy creating ones (*has_hypernym*, *has_meronym*, etc.) as well nonhierarchical ones (*near_synonym*, *be_in_state*, etc.). Using hierarchical connections makes WordNet to be a set of trees, whereby part of those trees are threaded (That is a fact from authors' analysis). The vertices of trees are synsets and edges are always some semantic connections.

Such tree has always a notion (synset) on the highest level (so-called root vertice) and other vertices on different levels. In given context we call root vertice also a root synset.

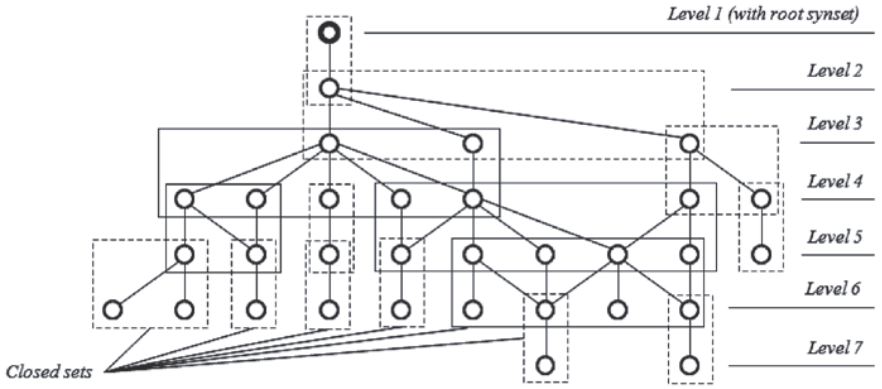


Figure 1. Natural tree of the WordNet with closed sets

[4] Results tables relevant to 'Anomalies in the WordNet Verb Hierarchy', paper delivered to COLING 2008, Manchester, UK, August 2008: http://www.rockhouse.me.uk/Linguistics/ (08.01.2013).

We have an invented example of such a WordNet tree in Figure 1. The synsets of the given tree (vertices) can be divided into seven levels. On the first level is the most general semantic synset – the root synset, and on the last levels (level 6 and level 7) synsets with a possibly concrete meaning. For example, based on semantic connection *has_hyponym* Princeton WordNet (version 3.0) has 346 root synsets (= trees) and Estonian WordNet (version 64) 204 synsets.

In order to understand closed subsets (Lohk et al. 2012b) in Figure 1 we have to consider only any two neighbouring levels. Let us take for example levels 3 and 4. If we separate those levels with their vertices, one can see that the connections between vertices create two closed sets of vertices. To recognise possible errors it is important to study such sets, where subsynset has a semantic connection with at least two different supersynsets. Such sets are presented in Figure 1 with thick lines and there are four of them. (The number of all closed subsets in Figure1 is 15). For example, Estonian WordNet (version 64) has as a maximal closed set with dimensions 4,945 x 457. In the language of Figure 1, this closed set has 4,945 vertices in the lower level and 457 vertices in the upper level.

The following table presents an overview of maximal closed sets in the WordNet-type dictionaries that we have analysed to date.

Table 1. Dimensions of the maximal closed set in a WordNet

Seq. No.	Name and version of the WordNet	Number of the synsets	Dimensions of the maximal closed set
1	Polish WordNet 1.7	105 074	28 279 x 3 595
2	Cornetto, 1.3	70 492	10 418 x 556
3	Estonian WordNet, 64	54 078	4 945 x 457
4	Princeton WordNet, 3.0	117 659	1 333 x 167
5	Finnish WordNet, 1.1.2	117 659	1 248 x 165
6	Catalan WordNet, 3.0	99 253	1 007 x 91
7	Slovenian WordNet, 3.0	42 919	248 x 3

The number of closed subsets separated using the semantic relation *has_hyperonym* for all those WordNets remains between 4000 and 20 000.

A very suitable algorithm to separate closed subsets is given by Flannery et al. (2009). An example of a closed subset with real data is presented in Figure 2.

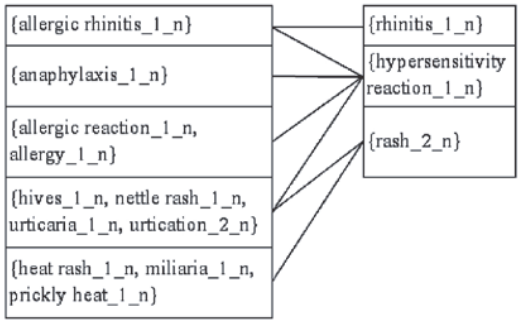


Figure 2. Real example of a closed subset (Princeton WordNet, version 3.0), rotated 90 degrees

The next section is dedicated to the study of such maximal closed sets.

3. Improving identification of mistakes by reducing the number of crossings

To visually identify possible mistakes in the connections between the synsets of a closed subset of a WordNet it is necessary to visualize the connections as clearly and with as little clutter as possible. One way to achieve this goal is to reduce the number of crossings in the graph representation of the WordNet by reordering vertices as shown in Figure 3.

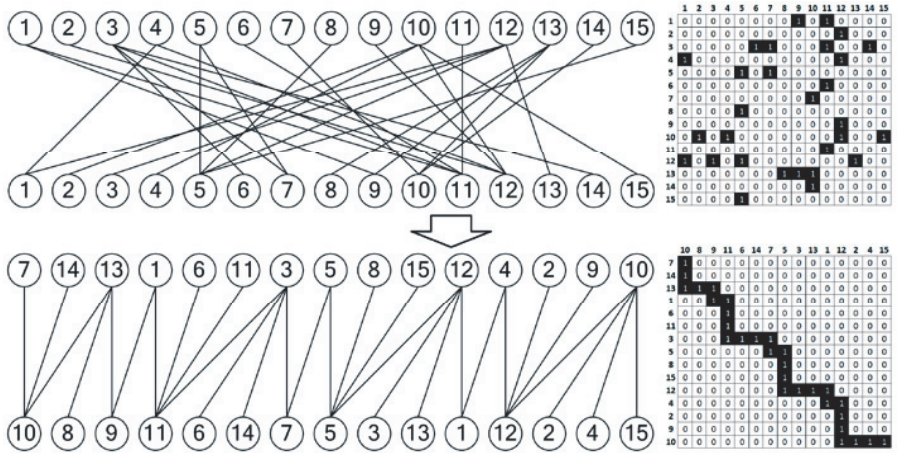


Figure 3. Graph (its corresponding adjacency matrix plotted on the right) with different permutations of vertices, illustrating how a good reordering can reduce the number of crossings in the bipartite graph (from 206 to 0 in this example)

There are many algorithms for this task, with different approaches – such as genetic algorithms (Mäkinen, Sieranta 1994), heuristic algorithms (e.g. barycenter (Sugiyama et al. 1981), median (Eades, Wormald 1994)) and for small graphs even exact methods (Jünger, Mutzel 1997). In the same paper in which Jünger and Mutzel introduced their exact method, they also compared different heuristic algorithms on larger graphs for which the exact method is not viable. They concluded that the iterated barycenter method was clearly the best choice for both its speed and solution quality.

3.1. The two-step method for reducing the number of crossings

In this work we introduce a novel technique which outperforms other widely used methods including barycenter heuristic. Our method consists of two steps:

1. Power iteration seriation;
2. Median heuristic.

First let us focus on the second step – the median heuristic by Eades and Wormald (1994). The median heuristic is a well known method for crossing minimization. To

reduce crossings, this method finds the median of the positions of adjacent vertices for every vertex and then ranks the vertices in a layer according to these values. Technically it is very similar to the barycenter heuristic (Sugiyama et al. 1981), the only difference being that the latter uses mean values of the positions of adjacent vertices instead of median values. This method is often applied iteratively, fixing one layer and reordering the other in turns, until there is no change in the order of vertices. The final outcome of the median (and also barycenter) heuristic depends on the initial state of the graph. To gain better results one can restart the algorithm a number of times with different random initial orderings and choose the best result but, as Jünger and Mutzel (1997) concluded, for bigger graphs the results improve only slightly. The purpose of the first step of our method is not to rely on random ordering, but to preprocess the graph with the aim of providing as good a starting point for the median heuristic as possible.

The first step of our method is a custom modification of a very fast (approximately linear to the input size), effective and simple clustering algorithm, Power Iteration Clustering (PIC) by Lin and Cohen (2010). The name of their method comes from the power iteration eigenvalue algorithm on which it is based.

The power iteration algorithm is used to find the dominant eigenvalue λ_1 (assuming there is one i.e. $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n|$) and eigenvector $\boldsymbol{v}_1$ of a matrix A. The algorithm takes the steps described in Figure 4. After a sufficient amount of iterations $\boldsymbol{b}_t$ converges to $\boldsymbol{v}_1$ of A.

Procedure Power Iteration (A):

1. Choose random nonzero vector $\boldsymbol{b}_0 \neq \mathbf{0}$; $t = 0$;
2. Calculate $\boldsymbol{b}_{t+1} = \frac{A\boldsymbol{b}_t}{\|A\boldsymbol{b}_t\|}$;
3. $t = t + 1$ and repeat step 2.

Figure 4. Description of power iteration eigenvalue algorithm

The PIC algorithm applies power iteration to a row-normalised (all elements in a row sum up to 1) similarity matrix W. Since the dominant eigenvector of W is a constant vector, it is useless for clustering (that's also the reason for additional constraint: $\boldsymbol{b}_0 \neq c\mathbf{1}$, i.e. initial vector must not be a constant vector). Therefore, to turn the power iteration eigenvalue algorithm into a clustering algorithm, Lin and Cohen augmented it with a stopping criterion which stops the process before converging to the constant dominant eigenvector. As a result we get a vector $\boldsymbol{b}_t$ (PIC-vector) which is an eigenvalue-weighted linear combination of all the eigenvectors of W and turns out to be a good clustering indicator. The main procedure of PIC is described in Figure 5, where $\boldsymbol{\epsilon}$ is a small number (e.g. 10^{-5}) used as a parameter for stopping criterion and $\boldsymbol{d}_t$ is a vector describing the changes (compared to previous iteration) in the values of the elements of vector $\boldsymbol{b}_t$. The algorithm is stopped when for two consecutive iterations $\boldsymbol{d}_t$ has remained almost constant i.e. none of the absolute differences of changes are larger than $\boldsymbol{\epsilon}$. Lin and Cohen used *k-means* on the PIC-vector to obtain the final result in the form of clusters.

Procedure $PIC(W, \epsilon)$:

1. Choose random vector $\boldsymbol{b}_0 \neq \mathbf{0} \wedge \boldsymbol{b}_0 \neq c\mathbf{1}$; $\boldsymbol{d}_0 = \mathbf{1}$
2. Calculate $\boldsymbol{b}_{t+1} = \frac{W\boldsymbol{b}_t}{\|W\boldsymbol{b}_t\|_1}$; $\boldsymbol{d}_{t+1} = |\boldsymbol{b}_{t+1} - \boldsymbol{b}_t|$;
3. If $\|\boldsymbol{d}_{t+1} - \boldsymbol{d}_t\|_\infty > \epsilon$, then $t = t + 1$ and repeat step 2; otherwise output $\boldsymbol{b}_t$.

Figure 5. Description of the main subroutine of PIC algorithm

Our own work has shown that PIC-vector can also be successfully used for seriation. To do that, we first calculate two PIC-vectors – one for rows and the other for columns. Then we reorder the rows and columns of the matrix according to the ascending or descending order of the values in the corresponding PIC-vector. The exact procedure is shown in Figure 6 where W_r and W_c are normalised similarity matrices, $\boldsymbol{b}_r$ and $\boldsymbol{b}_c$ PIC-vectors and $\boldsymbol{l}_r$ and $\boldsymbol{l}_c$ labels for reordering. Lower indices r and c denote rows and columns respectively.

Procedure $PISeriation(W_r, W_c, \epsilon)$:

1.	$\boldsymbol{b}_r = PIC(W_r, \epsilon)$	Get PIC-vector for rows...
2.	$\boldsymbol{b}_c = PIC(W_c, \epsilon)$	...and columns.
3.	$\boldsymbol{l}_r = sort(\boldsymbol{b}_r)$	Sort both vectors and get labels
4.	$\boldsymbol{l}_c = sort(\boldsymbol{b}_c)$	indicating the original positions.
5.	$A = A[\boldsymbol{l}_r, \boldsymbol{l}_c]$	Reorder rows and columns of A

Figure 6. Description of seriation procedure using PIC algorithm

For the first step of the crossing minimisation method we use the power iteration seriation with a very simple symmetric similarity function where the similarity $s(x_i,x_j) = s(x_j,x_i)$ between two vertices x_i and x_j from the same layer is equal to the number of their common neighbours in the opposite layer: $s(x_i,x_j) = s(x_j,x_i) = |n(x_i) \cap n(x_j)|$ (where $n(x)$ denotes the set of neighbours of x). If we represent the bipartite graph as an adjacency matrix A and nth row of A as A(n), then we can rewrite the function as follows:

$$s(x_i, x_j) = s(x_j, x_i) = \begin{cases} A(i) \cdot A(j), & \text{if } x_i \text{ and } x_j \text{ are upper layer vertices} \\ A^T(i) \cdot A^T(j), & \text{if } x_i \text{ and } x_j \text{ are lower layer vertices} \end{cases}$$

The similarity matrix of upper layer vertices S_r (or row similarity matrix of A) where element $S_r(i,j) = s(x_i,x_j)$ can then be calculated as $S_r = AA^T$ and the similarity matrix of lower layer vertices (or column similarity matrix of A) can be calculated as $S_c = A^TA$. Both matrices have to be normalised before using power iteration seriation on them.

Since the elements of PIC-vector corresponding to similar objects (vertices in our case) tend to obtain similar values, the positions of vertices after seriation also tend to correlate with the number of common neighbours. As a result subsets of vertices with many common neighbours clump together after processing with power iteration seriation in the first step of our method. This kind of approach alone does not always provide very good results in terms of the number of crossings. For example, it is possible that one layer has to be reversed, because power iteration seriation can produce results where the band of ones in the adjacency matrix runs from top-right

to bottom-left (Figure 7b) instead of top-left to bottom-right (Figure 7a), which is not good from the crossing number perspective. In some conditions (when the graph consists of more than one connected component or even when there are multiple components which are weakly connected to each other; with 'noise'; suboptimal ϵ, etc.) it is possible that some subset of similar vertices will be positioned too far away from their common neighbours (Figure 7c). Additionally, there is a risk that some subsets of vertices within layers could be in reverse order (Figure 7d). Even worse, very often multiple problems come up simultaneously.

All the problems mentioned above are solved by applying a median heuristic to the result of power iteration seriation. The median heuristic is not just compensating for the weaknesses of power iteration seriation, but the output of the latter is also a very good initial permutation for the former, enabling it to achieve much better results than some random permutation would. For example: on the inital graph from Figure 1, the iterative median heuristic could only reduce the number of crossings to 27 (Figure 8), while power iteration seriation in conjunction with median heuristic reduced the number of crossings to 0 (Barycenter heuristic reduced the number of crossings to 28).

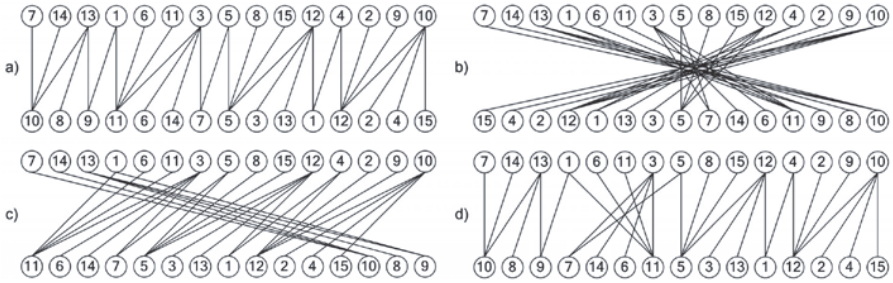


Figure 7. Example graph from Figure 1 illustrating some problems with power iteration seriation: a) one of the optimal permutations of vertices; b) lower layer in reverse order; c) subset of similar vertices in one layer are too far from their common neighbours; d) Subset of vertices in one layer is in reverse order

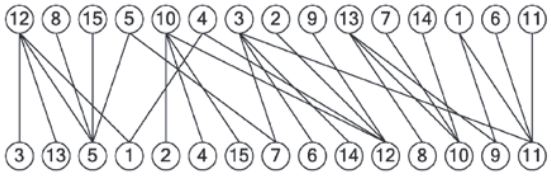


Figure 8. Result of iterative median heuristic on the initial graph from Figure 3

This kind of two-step method not only provides a much smaller number of crossings but may also provide these results while being faster than the iterative barycenter or median algorithm. This is possible because after preprocessing the graph with power iteration seriation, only one iteration of the median heuristic is sufficient to produce a superior result than the iterative barycenter or median method alone. If time is not crucial, then additional iterations of the median heuristic may be applied to polish the result further. Some additional improvement can also be found by trying different values for PIC's stopping criterion parameter ϵ (10^{-5} – 10^{-7} divided by

number of rows in similarity matrix was usually optimal for us). Next we will give some examples how this method performed on the real WordNet graphs.

3.2. Experiments on WordNet graphs

In this section we give an overview of our tests on the largest closed sets of synsets from Estonian and Princeton WordNets. The largest closed set from Estonian WordNet can be represented as a 4,945 by 457 matrix (see Table 1). In the case of the Princeton WordNet the matrix size is 1 333 x 167 (Table 1).

We ran our tests on a PC with 6 GB of RAM and an Intel® Core™ i7-870 Processor and compared three different methods: iterative barycenter, iterative median and our two-step method. In the two-step method we used only one iteration of median heuristic and for PIC's stopping criterion parameter ϵ we chose 10^{-5} divided by number of rows in similarity matrix. All 3 methods were run on the same initial permutation of vertices. The results are shown in Table 2.

Table 2. Results of three methods compared with a random permutation on the largest closed sets of two WordNets

	Estonian WordNet (v 64)		Princeton WordNet (v 3.0)	
	Time (s)	**Crossings**	**Time (s)**	**Crossings**
Initial	–	2 349 957	–	265 940
Median	4.1	904 629	0.9	36 862
Barycenter	16.9	308 444	1.5	22 927
2-step method	0.4	84 884	0.1	5 484

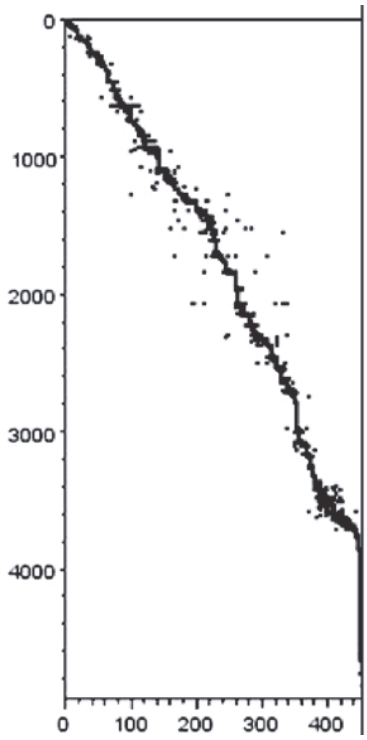
The two-step method turned out to be roughly 9–42 times faster than compared methods while producing more than 3–10 times fewer crossings. From these results we can conclude that our two-step method is the best choice for minimising the number of crossings in WordNet graphs.

Some possible ways of using the Minimal Crossing method to detect inconsistencies in WordNet structures is given by the author and others (Lohk et al. 2012a, 2012b). The detailed handling of those and others inconsistencies would require a separate article.

4. Looking at the results

As a result of using our two-step method we did get an ordered matrix (Figure 9a).

By converting this matrix with the labels of synsets into a MS Excel worksheet we have the possibility of studying the large closed subset more methodically. To make it easier to understand the result it is useful to freeze the headings of rows and columns. That makes it possible to move around in the table so that the synsets on both levels are always visible. To find possible errors one has to study such places in that table where conceptual synsets in rows and columns are conspicuously different. Usually such an occasion happens when one concept has several parents. The decision about a possible error will be naturally made by the lexicographer.



	{ehituskeraamika_1_n}	{katusekivi_1_n}	{läbiviik_1_n}	{teivas_1_n}	{ujuvmärk_1_n}	{ujuk_2_n}
{reimar_1_n}	0	0	0	1	1	
{tooder_1_n}	0	0	0	1	1	
{keraamiline katusekivi_1_n}	1	1	0	0	0	
{savikatusekivi_1_n}	1	1	0	0	0	
{savikivi_1_n}	1	1	0	0	0	
{savist katusekivi_1_n}	1	1	0	0	0	
{betoonkatusekivi_1_n}	0	1	0	0	0	
{harjakivi_1_n}	0	1	0	0	0	
{imbrex_1_n}	0	1	0	0	0	
{valtskivi_1_n}	0	1	0	0	0	
{antenni läbiviik_1_n}	0	1	1	0	0	
{antennikivi_1_n}	0	1	1	0	0	
{tuulutuskivi_1_n}	0	1	1	0	0	
{ventilatsiooni tuulutuskivi_1_n}	0	1	1	0	0	
{poi_1_n}	0	0	0	0	1	
{kork_2_n}	0	0	0	0	0	
{logilaud_1_n}	0	0	0	0	0	
{paravan_1_n}	0	0	0	0	0	
{pontoon_1_n}	0	0	0	0	0	
{ujuk_1_n}	0	0	0	0	0	

a) b)

Figure 9. The biggest closed subset of Estonia WordNet: a) closed subset after ordering; b) closed subset for the investigation, converted for Excel

5. Conclusion

Wordnet as a lexical and semantical database is widely used in different language technology applications. Therefore it is important to ensure the quality of any Wordnet used. Previous study has shown that WordNet with its hierarchical structures consists of many relations which quite easily cause errors in the Wordnet structure (Lohk et al. 2012). In this paper we propose a formal way to detect and study possible inconsistencies using closed subsets. The notion of a closed subset has been explained using the WordNet tree. Separated closed subsets are represented as matrices and a new and fast two-step method reorders such sparse relational systems into an easily visible and understandable view. Our method has been compared with other fast reordering methods and tested on Estonian and Princeton WordNets. As a final suggestion we transform the subsets with correct syntactic labels into an Excel spreadsheet to enable convenient study of places where the structural connections of concepts (synonym synsets) are suspicious.

References

Eades, Peter; Wormald, Nicholas C. 1994. Edge crossings in drawings of bipartite graphs. – Algorithmica, 379–403.

Flannery, P. B.; Press, H. W.; Teukolsky, A. S.; Vetterling, T. W. 2009. Numerical Recipes in C. The Art of Scientific Computing. South Asia: Cambridge University Press India.

Jing, H. 1998. Usage of WordNet in natural language generation. – Proceedings of the Workshop Usage of WordNet in Natural Language Processing Systems: COLING-ACL 1998; August 16, Montreal, Quebec, Canada, 128–134.

Jünger, Michael; Mutzel, Petra 1997. 2-Layer Straightline Crossing Minimization: Performance of exact and heuristic algorithms. – Journal of Graph Algorithms and Applications, 1–25.

Li, X.; Szpakowicz, S.; Matwin, S. 1995. A WordNet-based algorithm for word sense disambiguation. – Proceedings of IJCAI 1995. Morgan Kaufmann Publishers, 1368–1374.

Lin, Frank; Cohen, William W. 2010. Power iteration clustering. – Proceeding of the 27th International Conference on Machine Learning, June 21-24, 2010, Haifa, Israel. Omnipress, 655–662.

Liu, Y.; Jiangsheng, Y.; Zhengshan, W.; Shiwen, Y. 2004. Two kinds of hypernymy faults in Word-Net: the cases of ring and isolator. – Petr Sojka, Karel Pala, Pavel Smrz, Christine Fellbaum, Piek Vossen (Eds.). Proceedings of the Second Global WordNet Conference. Brno, Czech Republic, 20-23 January 2004. Masaryk University, 347–351.

Lohk, Ahti; Võhandu, Leo 2012. Eesti Wordnet'i struktuuri analüüsist. – Eesti Rakenduslingvistika Ühingu aastaraamat, 8, 139–151. http://dx.doi.org/10.5128/ERYa8.09

Lohk, Ahti; Vare, Kadri; Võhandu, Leo 2012a. First steps in checking and comparing Princeton WordNet and Estonian WordNet. – Miriam Butt et al. (Eds.). Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH. April 23-24 2012, Avignon, France. Association for Computational Linguistics, 25–29.

Lohk, Ahti; Vare, Kadri; Võhandu, Leo 2012b. Visual Study of Estonian WordNet using Bipartite Graphs and Minimal Crossing algorithm. – Proceedings of 6th International Global WordNet Conference, Matsue, Japan, 2012, 167–173.

Miller, G.; Beckwith, R.; Fellbaum, C.; Gross, D.; Miller, K. 1990. Introduction to WordNet: An on-line lexical database. – International Journal of Lexicography 3, 235–312.

Morato, J.; Marzal, M. Á.; Lloréns, J.; Moreiro, J. 2004. WordNet applications. – Petr Sojka, Karel Pala, Pavel Smrz, Christine Fellbaum, Piek Vossen (Eds.). Proceedings of the Second Global WordNet Conference. Brno, Czech Republic, 20-23 January 2004, 270–278.

Mäkinen, Erkki; Sieranta, Mika 1994. Genetic algorithms for drawing bipartite graphs. – International Journal of Computer Mathematics, 53 (3-4), 157–166. http://dx.doi.org/10.1080/00207169408804322

Orav, Heili; Kerner, Kadri; Parm, Sirli 2011. Eesti Wordnet'i hetkeseisust. – Keel ja Kirjandus, 2, 96–106.

Richens, Tom 2008. Anomalies in the WordNET verb hierarchy. – Proceedings of the 22nd International Conference on Computational Linguistics: COLING-ACL 2008, August, Manchester, UK, 729–736.

Rila, M.; Tokunaga, T.; Tanaka, H. 1998, The use of WordNet in information retrieval. – Proceedings of the Workshop Usage of WordNet in Natural Language Processing Systems: COLING-ACL 1998, August 16, Montreal, Quebec, Canada, 31–37.

Salam, Khan Md Anwarus; Khan, Mumit; Nishino, Tetsuro 2009. Example based English-Bengali machine translation using WordNet. – Proceedings of the Triangle Symposium on Advanced ICT 2009 (TriSAI 2009), October 28-30, 2009. Tokyo, Japan.

Sugiyama, Kozo; Tagawa, Shojiro; Toda, Mitsuhiko 1981. Methods for Visual Understanding of Hierarchical System Structures. – IEEE Transactions on Systems, Man and Cybernetics, 11 (2), 109–125. http://dx.doi.org/10.1109/TSMC.1981.4308636

Vider, Kadri 2001. Eesti keele tesaurus – teooria ja tegelikkus. – Margit Langemets (Toim.). Leksikograafiaseminar "Sõna tänapäeva maailmas" / Leksikografinen seminaari "Sanat nykymaailmassa". Ettekannete kogumik. Eesti Keele Instituudi toimetised 9. Tallinn: Eesti Keele Sihtasutus, 134–156.

Web References

The Global WordNet Association. http://www.globalwordnet.org/gwa/wordnet_table.html (08.01.2013).

Results tables relevant to "Anomalies in the WordNet Verb Hierarchy" paper delivered to COLING 2008. Manchester August 2008. http://www.rockhouse.me.uk/Linguistics/ (08.01.2013).

Princeton WordNet (version 3.0, 3.1). http://wordnet.princeton.edu/ (08.01.2013).

Estonian WordNet (version 65). http://www.cl.ut.ee/ressursid/teksaurus/test/estwn.cgi.et (08.01.2013).

Ahti Lohk (Tallinn University of Technology), main research interests are in the field of data analysis.
ahti.lohk@ttu.ee

Ottokar Tilk (Tallinn University of Technology), main research interests are data analysis and machine learning algorithms.
ottokar.tilk@ttu.ee

Leo Võhandu (Tallinn University of Technology), main research interests are in the field of data analysis.
leo.vohandu@ttu.ee

KUIDAS LUUA KORDA WORDNET'I TÜÜPI SÕNARAAMATUTE SUURTES KINNISTES ALAMHULKADES

Ahti Lohk, Ottokar Tilk, Leo Võhandu
Tallinna Tehnikaülikool

WordNet kui leksikaalsemantiline andmebaas leiab laialdast kasutust keeletehnoloogia rakendustes, mistõttu on ilmne, et tulemuse kvaliteet sõltub paljuski *wordnet*'i enda kvaliteedist. Varasemad uurimused on näidanud, et *wordnet*'i hierarhiat tekitavates puudes esineb seoseid, mis põhjustavad tema struktuuris vigu (Lohk, Võhandu 2012). Ühe võimalusena pakutakse artiklis taolisi kõrvalekaldeid uurida ja avastada kinniste alamhulkade kaudu, mida esitatakse maatriksina ja millele rakendatakse autorite pakutud uudset kahesammulist meetodit. Kinniseid alamhulki selgitati tehislikult koostatud *wordnet*'i puu alusel. Pakutud kahesammulist meetodit, mis sobib suurte relatsiooniliste süsteemide korrastamiseks, kõrvutati teiste kiirete varasemate meetoditega (raskuskeskme meetod ja mediaanmeetod). Jõuti järeldusele, et kahesammuline meetod pakub tulemuseks nii paremat ristumiste arvu kui ka kiiremat algoritmi kui varasemad meetodid. Meetodit testiti Eesti ja Princetoni *wordnet*'idel. Maatriksina saadud tulemusi soovitati koos sünohulkade nimedega konverteerida tabelarvutusprogrammi, liikuda mööda korrastatud maatriksil olevat lairiba ning uurida ridades ja veergudes olevaid sünohulkade neid kohti, kus mõisted silmatorkavalt erinevad.

Võtmesõnad: tesaurus, suletud hulgad, järjestamine, klasterdamine iteratiivse astendamisega, ristumiste arvu vähendamine, WordNet

APPENDIX C

Lohk, Ahti; Allik, Kaarel; Orav, Heili; Võhandu, Leo (2014). Dense Components in the Structure of WordNet. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14): LREC2014, Reykjavik, Iceland, May 26-31, 2014. (Edit.) Nicoletta Calzolari and Khalid Choukri and Thierry Declerck and Hrafn Loftsson and Bente Maegaard an. ELRA, 2014, pp. 1135 - 1139.

Dense Component in The Structure of Wordnet

Ahti Lohk[1], Heili Orav[2], Kaarel Allik[1], Leo Võhandu[1]

Tallinn University of Technology[1], University of Tartu[2]
Akadeemia tee 15a, Tallinn, Estonia[1]
Liivi 2, Tartu, Estonia[2]
ahti.lohk@ttu.ee, heili.orav@ut.ee, kaarel.allik@ttu.ee, leo.vohandu@ttu.ee

Abstract

This paper introduces a test-pattern named a dense component for checking inconsistencies in the hierarchical structure of a wordnet. Dense component (viewed as substructure) points out the cases of regular polysemy in the context of multiple inheritance. Definition of the regular polysemy is redefined – instead of lexical units there are used lexical concepts (synsets). All dense components are evaluated by expert lexicographer. Based on this experiment we give an overview of the inconsistencies which the test-pattern helps to detect. Special attention is turned to all different kind of corrections made by lexicographer. Authors of this paper find that the greatest benefit of the use of dense components is helping to detect if the regular polysemy is justified or not. In-depth analysis has been performed for Estonian Wordnet Version 66. Some comparative figures are also given for the Estonian Wordnet (EstWN) Version 67 and Princeton WordNet (PrWN) Version 3.1. Analysing hierarchies only hypernym-relations are used.

Keywords: wordnet, test-pattern, dense component

1. Introduction

Wordnet (Miller and Fellbaum, 1998) as a lexical resource is attractive due to its hierarchical structure of synonym sets (synsets), which is helpful for many natural language processing (NLP) tasks. Wordnet is mostly used for machine translation, automate analysis of text and word sense disambiguation, but also for text categorization, information retrieval, text mining and even for creating new wordnets (Morato et al., 2004). Polysemy as a feature of wordnet hierarchical structure may complicate the NLP (Veale, 2004) and affect the quality of these applications. At the same time, the polysemy may help to find and define new semantic relations between lexical units or synsets which in turn help to improve utility of wordnet in NLP tasks. (Barque et al., 2009) and (Freihat et al., 2013) use regular polysemy patterns to discover these new semantic relations. In our research we redefine the meaning of regular polysemy. To find the cases of regular polysemy in the hierarchical structure of wordnet we use a test-pattern named a *dense component* which is viewed as a substructure of the wordnet hierarchy. Generally defining, the *dense component* is a bipartite graph that has at least two synsets with at least two identical parents, but could contain additional synsets with some common parents (synsets with dotted line) as shown in Figure 1.

With respect to the state of the art, regular polysemy in wordnet is viewed as a status where at least two lexical units (members of synset) from the same or different level in hierarchical structure are related to the combination of one of the following:

1. lexical units (from higher level synsets) (Peters and Peters, 2000; Freihat et al., 2013);
2. "conceptual signposts"(Peters and Peters, 2000)[1];

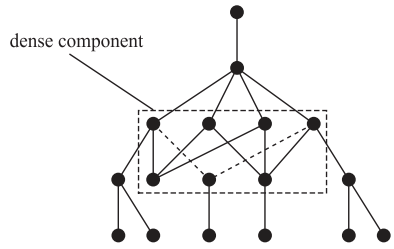


Figure 1: Dense component in a hierarchical structure

3. top ontology concepts, unique beginners or domain category names (Buitelaar, 1998; Freihat et al., 2013).

However, in our view we apply the same idea of regular polysemy (RP) but instead of abovementioned categorization we use synsets as lexical concepts. So redifining the RP we say that RP is a status where at least two synset have at least two hypernyms with similar relations between those hypernyms.

The paper fills the gap in the state-of-the art by asking the main research question of how to check and evaluate regular polysemy in the hierarchical structure of wordnet? To answer the question, we present a test-pattern named *dense component* view on substructures of the wordnet hierarchy in case of regular polysemy.

The structure of the paper is as follows: Section 2 gives additional background for understanding the main body of paper. Next, Section 3 presents formalized algorithm of dense component. Section discusses the inconsistencies taxonomy. Section 5 evaluates the dense component yielding a

[1]"A pair-wise combinations of nodes in the WordNet hierarchy that are preferably more specific than the unique beginners but still general enough to encompass several words and constitute semantically homogenous groups"

numerical overview and finally, Section 6 concludes the paper.

2. Features of Wordnet Dictionaries

Wordnets share properties for the concepts of polysemy that are part of the definitions of the test patterns. On the other hand, regular polysemy is only part of one test-pattern definition, namely the pattern *dense component*. In the remainder, Section 2.1. gives general structural features for wordnet and Section 2.2. polysemy versus regular polysemy.

2.1. Wordnet-like dictionaries

The fundamental approach for designing WordNet-like dictionaries came from Princeton WordNet (Miller, 1990). Each WordNet shares particular structural features. First, synonym sets (synsets) group many synonyms that share the same meaning and are also referred to as concepts. Semantic relations connect synsets to each other, e.g., by *hypernymy*, *meronymy* for creating a hierarchical structure, and *caused by*, *near synonym* that do not create a hierarchical structure. In this article, we consider only hypernymy-hyponymy relations as objects of analysis. Furthermore, there is no extension limitation for the approach to different semantic relations that shape the hierarchical structure. For details about Estonian Wordnet, we refer the reader to (Õim et al., 2010). Furthermore, Princeton WordNet has 117 773 synsets and 88 721 hypernym-hyponym relations. In Estonian Wordnet Versions 66, these values are 58 566 and 51 497 respectively, while for Versions 67, the values are 60 434 and 52 678 respectively. Princeton WordNet has hypernym-hyponym relations only in cases of nouns and verbs; in Estonian Wordnet in case of nouns, verbs and adjectives.

2.2. Regular polysemy

According to (Ravin and Leacock, 2000), polysemy is multiplicity of meanings of words. In wordnets, polysemy should appear as one concept with several hypernyms. If the latter are regularly included then the polysemy itself is regular. The best known definition of regular (also systematic or logic) polysemy gives (Apresjan, 1974). In (Langemets, 2010) Apresjan's definition is simplified: regular polysemy is a status where at least two words have at least two meanings with similar relation between those meanings. For example, if the word *school* has meanings *institution* and *building* than the same is true about a *hospital*. The latter is also an *institution* as well as a *building*. According to (Freihat et al., 2013), *institution-building* is an example for a polysemic pattern. Our goal in regular polysemy cases is to check if the polysemic pattern is justified with respect to regular polysemy.

Following section formulates a mathematical concept of dense component.

3. Definitions and algorithm of the dense component

Let $G = (Y, A, E)$ be a bipartite graph whose partition has the parts Y and A; $E \subseteq A \times Y$ is the set of edges. Let $|Y| = m$ and $|A| = n$.

Our goal is to glue together some nodes from A under certain conditions. Therefore it is convenient to represent the result by

$$\hat{G} = \{g_i : g_i =< L_i, N_i >; \quad i = 1, \ldots, k; 1 \leq k \leq n\},$$

where L_i is the set of glued nodes from A and N_i is the set of neighbours of L_i.

For a natural number τ we define a binary relation $R(\hat{G})_\tau \subseteq \hat{G} \times \hat{G}$:

$$R(\hat{G})_\tau = \{(g_i, g_j) : g_i \in \hat{G}, g_j \in \hat{G}, |N_i \cap N_j| \geq \tau\}.$$

We say, that g_i and g_j from $\hat{G}$ are τ-*connected*, if $(g_i, g_j) \in R(\hat{G})_\tau$. Obviously, $R(\hat{G})_\tau$ is symmetrical and reflexive and the emptyness of $R(\hat{G})_\tau$ can be detected in time $\mathcal{O}(k^2 \cdot m)$.

For given $\hat{G}$ and $(u, v) \in R(\hat{G})_\tau$ we denote

$$glue(u, v, R(\hat{G})_\tau) = (\hat{G} \backslash \{u, v\}) \cup z,$$

where $z =< L_u \cup L_v, N_u \cup N_v >$.

Algorithm 1 *τ-closure*
$l := 0$; $\hat{G}^0_\tau := \{< \{g_i\}, N_i >: g_i \in A, N_i = \{y : (g_i, y) \in E\}\}$;
while $R(\hat{G}^l)_\tau \neq \emptyset$
do choose $u, v : (u, v) \in R(\hat{G}^l)_\tau$;
$\hat{G}^{l+1}_\tau := glue(u, v, \hat{G}^l_\tau)$; $l := l + 1$;
od $\hat{G}^+_\tau := \hat{G}^l_\tau$;

The result of the algorithm, $\hat{G}^+_\tau$ is called τ-*closure* of G.

Every step of the cycle glues two nodes, therefore the Algorithm 1 halts after at most $n - 1$ steps.

Due to commutativity and associativity of the set union ($\cup$), the τ-closure does not depend on the order of choosing elements in the body of the cycle. Therefore $\hat{G}^+_\tau$ is unique for G.

Definition: Dense component is every item g in graph $\hat{G}^+_\tau$ (Algorithm 1) and it is corresponding to Fig. 1.

$$g \in \hat{G}^+_\tau \quad (2)$$

In next section we explain what kind of inconsitencies can be found with help of previously described algorithm of finding dense components.

4. Inconsistencies of Substructure

4.1. Inconsistency taxonomy

Inconstency types lexicographer is focusing on are following:

1. **Non-justified regular polysemy –** in accordance with Section 2.2., linguists have to check if the regular polysemy is justified or not. Furthermore, having expanded view of dense component (i.e. additional synsets connected to dense component), perspective of regular polysemy may help to detect situations where there exist other synsets that are not connected to the same polysemic pattern as shown in Figure 2.

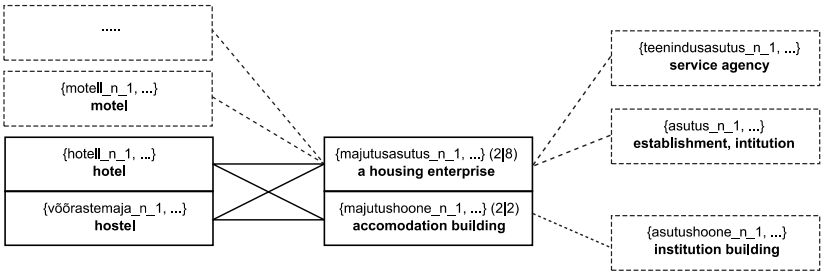


Figure 2: Dense component, non-regular use of polysemy

2. **Ignoring the principle of economy (redundant semantic relation) –** this inconsistency is typical to an asymmetric ring topology in cases where one branch is redundant such as in Figure 2. (Liu et al., 2004; Richens, 2008).

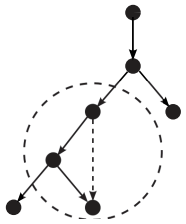


Figure 3: Asymmetric ring topology, dotted line is redundant

3. **Inappropriate semantic relationship –** it implies that a semantic relationship type must change. This inconsistency affects every test-pattern.

4. **Wrongly inherited domain category –** if one synset inherits simultaneously two different domain categories, one of them is wrong (Liu et al., 2004). The gloss of the synset indicates which of the categories is most appropriate (Miller and Fellbaum, 1998). Unfortunately, this inconsistency is applicable only on PrWN, because its every synset has the information about the domain category in contrast to EstWN.

4.2. Some examples

In this section we present three dense component examples with their specific inconsistencies. In order to facilitate the work of lexicographer, additional synsets connected to dense component are marked using dotted line. Mostly connected synsets (usually located in the middle of the figures 2, 4, 5) are called parents of the dense component. Every parent contains information about its number of subordinates (represented in brackets). First number shows connections to subordinates in the dense component, and second one refers to total number of subordinates (see Figures 2, 4, 5).

In **Figure 2**, we have typical case where the regular polysemy is allowed **–** *hotel* is simultaneously the building and the institution. While the nature of the *motel* is similar to the *hotel* we expect that the *motel* is connected to same polysemic pattern as the *hotel*.

In **Figure 4**, we see the case where the concept *cinnabar* mistakenly has got three hypernyms. According to the definition of *cinnabar*, only one hypernym was left for *cinnabar – mineral*. Colors as part of material have been changed to *holonym* instead of *hypernym*.

In **Figure 5**, we meet the asymmetric ring topology case. In order to facilitate the work of the lexicographer all these relvant synsets can be higlighted as shown in the case of *artistic production, art*. At the same time this is the case where one co-hypernym ({*artistic production, art*}) becomes to be parents for another ({*applied art*}). I.e., {*artistic production, art*} links with {*glasswork, ...*} and {*leatherwork, ...*} have to be removed.

5. Evaluation

In this section we compare EstWN Version 66 to 67 to see the changes that have taken place in wordnet hierarchy after correcting the dense components by the lexicographer. Hereby, we focus on four different changes as follows: the number of multiple inheritance, sizes of dense components, the number of dense components and distribution of errors.

5.1. The number of multiple inheritances

Every polysemic case in dense component is related to multiple inheritance, i.e. with synsets that have at least two parents/hypernyms in wordnet hierarchy. Therefore correcting a dence component it affects multiple inheritances as well.

Nr of parents	**EstWN, v66** (number of synsets)	**PrWN, v3.1** (number of synsets)	**EstWN, v67** (number of synsets)
5	1	1	–
4	5	3	1
3	68	30	32
2	1 603	1391	1 131
SUM	1 677	1 425	1 164

Table 1: Multiple inheritance counts before and after analysis and correction of the dense components.

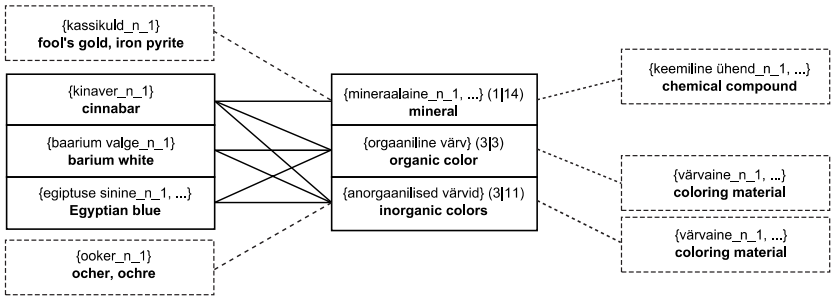


Figure 4: Dense component, wrong semantic relation

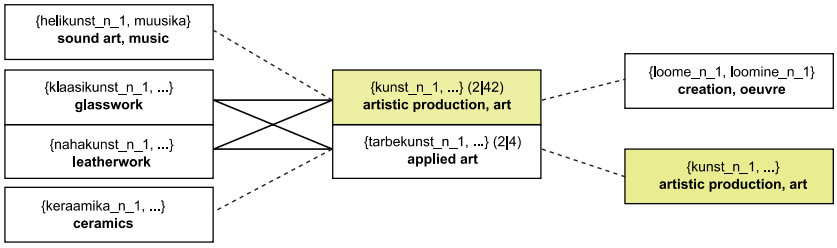


Figure 5: Dense component, asymmetric ring topology

Looking at the Table 1, we see that after correction of dense components there are no synsets with 5 parents in Version 67. Synsets with 3 parents are reduced about 50% and dual inheritance is reduced about by 500 cases.

5.2. Size and number of the dense components

According to the number of parents in dense components we present in Table 2 ten components with the highest number of parents with their occurances for two EstWN versions and for one PrWN version.

A considerable change after correction of dense components can be observed in their number of occurences. In the last row of Table 2 we see that the number of dense components is fallen from 121 to 24. The number of the biggest dense components (according to the number of parents) and the number of small dense components have also significantly decreased. E.g., both wordnets EstWN Version 66 and PrWN Version 3.1 include the same number of the smallest dense component (2 x 2) – 59. After correction this number dropped to 11.

Nr	**EstWN, v66** (synsets x parents) x nr	**PrWN, v3.1** (synsets x parents) x nr	**EstWN, v67** (synsets x parents) x nr
1	(5 x 9) x 1	(3 x 5) x 1	(3 x 3) x 1
2	(6 x 6) x 1	(2 x 5) x 1	(2 x 3) x 1
3	(116 x 4) x 1	(4 x 4) x 1	(8 x 2) x 1
4	(5 x 4) x 1	(3 x 4) x 1	(7 x 2) x 1
5	(3 x 4) x 1	(2 x 4) x 2	(6 x 2) x 1
6	(2 x 4) x 3	(9 x 3) x 1	(5 x 2) x 1
7	(19 x 3) x 1	(4 x 3) x 2	(4 x 2) x 2
8	(10 x 3) x 1	(3 x 3) x 3	(3 x 2) x 5
9	(8 x 3) x 2	(2 x 3) x 7	(2 x 2) x 11
10	(4 x 3) x 1	(9 x 2) x 2	–
SUM	121	107	24

Table 2: Dense components (bipartite graphs) sizes in EstWN (v66), PrWN (v3.1) and EstWN (v67). First ten components.

5.3. Distribution of corrections

In Table 3 we give a detailed overview about corrections that were made by the lexicographer. This table is based on comparing dense components from EstWN Version 66 to Version 67 manually. The sum of the first column numbers (106+14+65+39+14) in Table 3 is not equal to 121, because in many types of corrections have been included by the same dense components.

The number 106 presented in the first row points to the situation where dense component as a pattern is useful particularly in the checking of justness of regular polysemy cases. If regular polysemy is not justified, it means that some semantic relations have just been removed.

While asymmertic ring topology is possible in cases where direct link exceeds/overpasses more then one level of hierarchicy, we can not expect that dense component refers to all these kinds of inconstistencies.

In th third row, in about 50% of cases of dense components were engaged in the process of changing the semantic relations. Within this, 162 semantic relations were changed.

Hypernym relation was exchanged to near synonym 88 times, to fuzzynym 52 times etc.
Hierarchy was changed 39 times. Main reason was in circumstances where one co-hypernym or co-hypenym became parent to the another.
Only 14 dense components did not need any corrections. However, Version 67 consists of 24 dense components. These 24 had their content as follows:

- 14 of them were without any correction
- 2 of them were changed a little bit
- 8 of them were new

Futhermore, all dense components in Version 66 were revised, 1 868 synsets and 1 181 semantic relations were added into Version 67 as well. For that reason new 8 dense components arised in Version 67. regularity of multiple inheritance was not justified

106	regularity of multiple inheritance was not justified		
14	the principle of economy was not followed		
65	dense components was connected to changes of semantic relation		
	162	semantic relation was changed to	
		88	near synonym
		52	fuzzynym
		20	holonym
		2	meronym
39	hierarchy was changed in cases		
	14	co-hypernyms/co-hyponyms, one became parents to other one	
	7	connection to a synset is replaced to other one	
	5	new synset was added	
	4	added or removed lexical units from synsets	
	3	synsets were merged	
	2	removed synsets	
	4	hierarchical strcuture was reorganized	
14	no correction needed		

Table 3: Distribution of corrections

6. Conclusion

In this paper, we propose to use a dense component as a test-pattern to detect inconsistencies in substructures of wordnet hierarchy. Dense component is viewed on the one hand as bipartite graph and on the other hand as substructure of wordnet hierarchy and as a visual picture. It consists of at least one regular polysemy and simultaneously at least two synsets with at least two identical parents. Its finding process takes place iteratively trying to find fore current dense component other synsets that have at least two parents among the current dense component (see Section 3). The greatest benefit the dense component may give to lexicographer is helping to check the correctness of regular polysemy, i.e., it helps to see if the regular polysemy is justified or not but it is not limited to that case. Exhaustive analysis made by second author surprised positively because only 12% of dense components did not need any correction. The number of dense components in EstWN Version 66 diminished after corrections from 121 to 24 in Version 67.

7. Acknowledgements

In this paper Ahti Lohk is supported by Estonian National Doctoral School in Information and Communication Technology.

8. References

J. D. Apresjan. 1974. Regular polysemy. *Linguistics*, 12(142):5–32.

L. Barque, F.-R. Chaumartin, et al. 2009. Regular polysemy in wordnet. *JLCL-Journal for Language Technology and Computational Linguistics*, 24(2):5–18.

P. Buitelaar. 1998. *CoreLex: systematic polysemy and underspecification*. Ph.D. thesis.

A. A. Freihat, F. Giunchiglia, and B. Dutta. 2013. Approaching regular polysemy in wordnet. In *eKNOW 2013, The Fifth International Conference on Information, Process, and Knowledge Management*, pages 63–69.

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

Y. Liu, J. Yu, Z. Wen, and S. Yu. 2004. Two kinds of hypernymy faults in wordnet: the cases of ring and isolator. In *Proceedings of the Second Global WordNet Conference*, pages 347–351.

G. Miller and C. Fellbaum. 1998. *Wordnet: An electronic lexical database*. MIT Press Cambridge.

J. Morato, M. A. Marzal, J. Lloréns, and J. Moreiro. 2004. Wordnet applications. In *GLOBAL WORDNET CONFERENCE*, volume 2, pages 270–278.

H. Õim, H. Orav, K. Kerner, and N. Kahusk. 2010. Main trends in semantic-research of estonian language technology. In *Baltic HLT*, pages 201–207.

W. Peters and I. Peters. 2000. Lexicalised systematic polysemy in wordnet. In *LREC*.

Y. Ravin and C. Leacock. 2000. *Polysemy: Theoretical and computational approaches*. MIT Press.

T. Richens. 2008. Anomalies in the wordnet verb hierarchy. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 729–736. Association for Computational Linguistics.

T. Veale. 2004. Polysemy and category structure in wordnet: An evidential approach. In *LREC*.

APPENDIX D

Lohk, A.; Võhandu, L. (2014). Independent Interactive Testing of Interactive Relational Systems. A. Gruca, T. Czachórski, S. Kozielski (Toim.). Man-Machine Interactions 3 (63 - 70). Springer

Independent Interactive Testing of Interactive Relational Systems

Ahti Lohk and Leo Võhandu

Abstract Many ontologies and WordNet type dictionaries have been created with different interactive tools using specalists semantic knowledge to model complicated relational structures. All that type man-machine systems have usually many inconsistencies which are not easy to find. In our article we present a formal methodology which analyzes and tests any given relational table and hands out with a high probability erroneus subpatterns. The final testing and estimating will again be made by a specialist. While looking for structural and lexicographic errors we present as an example test results for Princeton WordNet (PrWN, version 3.1) and Estonian WordNet (EstWN, version 65) and a new substructure found in WordNets possibly pointing to the errors in the WordNet structure.

Key words: subpatterns, structural and lexicographic errors, WordNet type dictionaries, WordNet structure

1 Introduction and Background

The main idea and basic design of all WordNet type dictionaries came from Princeton WordNet [10]. Each WordNet is structured along the same lines: synonyms (sharing the same meaning) are grouped into synonym sets (synsets). Synsets are connected to each other by semantic relations, like "hypernymy" and "meronymy" (creating hierarchical structure) and "is caused by" and "near synonym" (creating non-hierarchical). In this article only

Ahti Lohk
Department of Informatics, Akadeemia tee 15a, Tallinn, Estonia, e-mail: ahti.lohk@ttu.ee

Leo Võhandu
Department of Informatics, Akadeemia tee 15a, Tallinn, Estonia, e-mail: leo.vohandu@ttu.ee

hypernymy-hyponymy relations are considered as objects of analysis. Of course, it is easy to extend the analysis over different word classes and different semantic relations.

Description of Estonian WordNet and its properties has been given by Orav et al. [11].

The study of WordNet type dictionaries structure is necessary to get feedback from the system. Some feedback is given by programs which help to create and maintain such dictionaries (e,g, Polaris [9], DEBVisDic [4],WordNetLoom [12], sloWTool [2], etc). Additionally there are some applications which help open the concepts context (Visual Tesaurus [15], Visuwords [7], Snappy Words [1] etc). Those applications allow one to find mistakes mainly by chance, there exist no summary reports or lists of error structures. From our viewpoint such reports are a must and they have to be analyzed by lexicographers. So we have a situation, where computer programs (created by authors) find the substructures pointing to errors and a specialist - lexicographer estimates the origin of possible errors.

Numerically, PrincetonWordNet has 117,773 synsets and 88,721 hypernymhyponym relations. In Estonian WordNet are these values respectively 56,928 and 49,181. Princeton WordNet has hypernym-hyponym relations only in case of nouns and verbs, in Estonian WordNet in case of nouns, verbs and adjectives.

In the next chapter we bring a short survey of substructures pointing to possible errors. In the chapter 3 authors suggest a new substructure which characterizes dictionaries of Wordnet type and points to a possible error origin.

2 Related Works

Twenty-seven tests for validating WordNets are proposed by Smrž [15]. Most of them are editing errors like *"empty ID, POS, SYNONYM, SENSE (XML validation)" or "duplicate literals in one synset"*. But there are also some tests for errors of hierarchical structure like: *"cycles", "dangling uplinks", "structural difference from PWN and other wordnets", "multi-paren relations"*. Liu et al [6] have found two cases that should be handled during the evolution of WordNet - rings and isolators. Richens [13] (referencing to work of Liu), has developed the idea of Liu's rings distinguishing two type of rings:

- **Asymmetric ring topology** (Fig 1, substructure nr "1")
- **Symmetric ring topology** (Fig 1, substructure nr "2")

Both Smrž [14] and Richens [13] have emphazised that problem of rings in the WordNet structure is caused (at least in part) by a situation, where one Independent Interactive Testing of Interactive Relational Systems 3 concept has several parents. Vider [16] arms this opinion and asserts that in ideal case

one concept has only one parent (hypernym). A detailed survey of cycles (Fig 1, substructure nr "3") has been given by Levary et al [5] with analyzes of Princeton WordNet 2.0. Clear understanding of that problems essence has already created generally a situation, where in WordNet type dictionaries there are no cycles or there are few cases (see Sect. **??**).

Next three subsections will give an overview about three substructures representing the newest results of Princeton and Estonian WordNet.

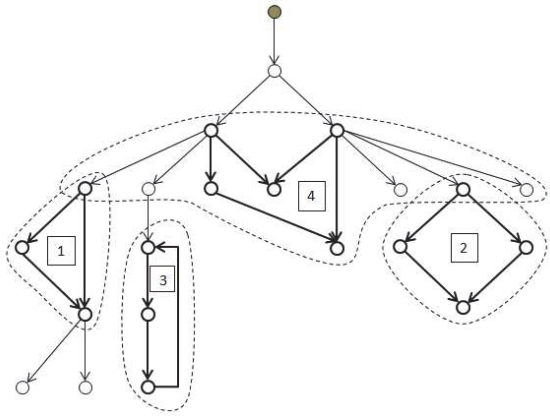


Fig. 1 Substructures in wordnet structure (artificially constructed tree).

2.1 Asymmetric ring topology

Many synset related semantic connections represent the situation where lexicographers have designated a new, more precise link to another synset, but did forget to remove the previous relation. In this case one synset is connected to hypernym-synsets twice - directly and indirectly through other hypernymsynset. This type of error occurs most frequently in the case of EstWN where redundant links appear 108 times. In PrWN there are 24 redundant links of hyponym.

In Figure 2 a redundant link is represented as a dotted line.

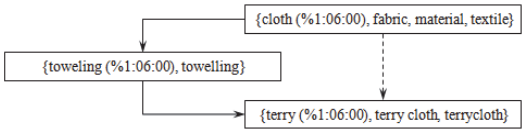


Fig. 2 Asymmetric topology ring, hyponym relation, PrWN

2.2 Symmetric ring topology

Liu et al [6] and Richens [13] describe a substructure which has according to Figure 1 substructure number 2. Liu et al [6] have an opinion that if two hyponyms of a single hypernym exist then they must have opposite properties in some dimension and hence cannot have a common hyponym, because a hyponym must inherit all the properties of its hypernym. They also argue that this problem emerges when both hypernyms are in same category. According to this base in Figure 3 an exception is shown where synset/concept {carrier(%1:18:00), immune carrier} with category/domain 18 has one parent from the same category ({immune (%1:18:00)}) (according to Liu *"main category"*) and other from ({transmitter (%1:17:00::), vector} (*"the less important category"*).

Exactly the same substructure as shown in Figure 3 does not exist in PrWN (version 3.1) and EstWN (version 65). So, we found only cases where between the highest concept ({causal agency (%1:03:00::), ...}) and the lowest concept ({carrier (%1:18:00::), immune carrier}) we have more than one level of concepts (see Figure 3).

Symmetric ring topology occured in the case of PrWN 215 times and in the case of EstWN 173 times.

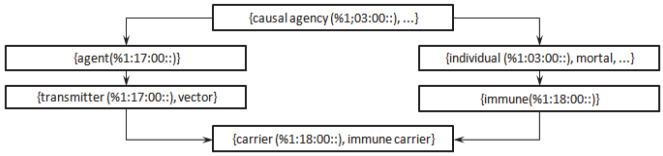


Fig. 3 Symmetric ring topology, hyponym relation, PrWN

Given subconstruction is not limited to two level subordinates but there can be even more levels of subordinates as it is shown in Figure 2. Symmetric ring topology occured in the case of PrWN 215 times and in the case of EstWN 173 times.

2.3 Cycles

A cycle can be discovered, by traversing the nodes of the hierarchical structure and reaching the same node repeatedly. Cycles are a very rare phenomenon in the Princeton WordNet. Only one cycle has been discovered in the case of semantic relation of domain category (see Figure 3). Our analysis of Estonian Wordnet last versions has shown that they do not have cycles any more.

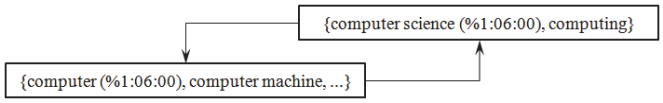


Fig. 4 Cycle, semantic relation domain category, PrWN

All those three substructures pointing to a possible error need the intervention of a lexicographer. In the first case (section 2.1) and third case (section 2.3) of substructures it can be sufficient to delete the superfluous connection. In the case 3.2 there could be a need for a more fundamental refreshening process.

3 The heart-shaped substructure

A special substructure has been found in the wordnet hierachical structure. Two concepts (hypernym-synsets) are related through a subconcept (subordinate) directly and through another subconcept – one concept directly, other concept indirectly. The adjacent picture shows that two hypernym-synsets share same subconcepts twice. Present construction needs some explanations about its usefulness. But, the first observations made personally by Fellbaum (personal communication, January 17, 2013) have given a positive feedback. All viewed images have been pointing to some errors in the structure. Not once was that structure detected in the case of verbs and hyponym relation. Noun and hyponym relation occurred 149 times.

3.1 An Example

There has been a hypothesis proposed by Fellbaum: many synsets with multiple inheritance may not be an error but reflect two different hyponym relations, such as type and role. One of the main techniques to define or check semantic relations Role or Type is to use "rigidity" attribute/property [3].

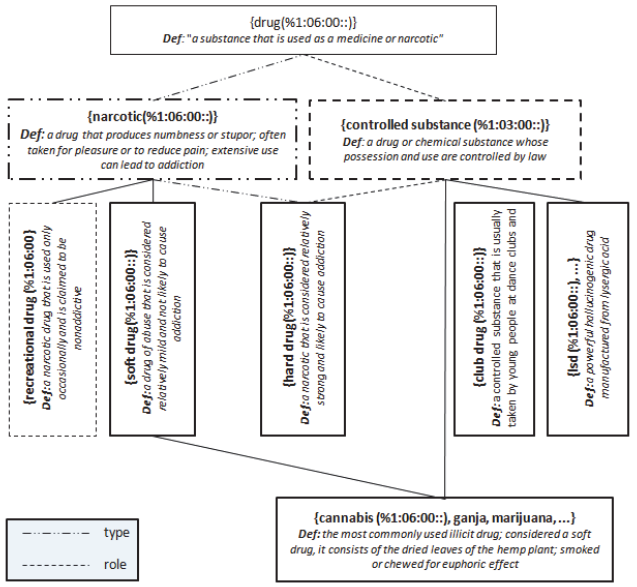


Fig. 5 Heart-shaped substructure, hyponym relation, PrWN

The main idea is that, if a superconcept (synset) is a rigid concept then the semantic relation should be role, but when the superconcept is a non-rigid concept then the semantic relation should be type. In order to check these kind of relations one can ask:

1. Is X always or necessarily a Y?
2. Can X stop being a Y?

If the answer to the first question is *"yes"* or to the second one *"no"* then the semantic relation should be type, but in the opposite case role. Figure 5 shows a heart-shaped substructure with hyponym-relations. The drugs/medicines can be linked to a Type superordinate(and give its chemical properties) and to a Role superordinate (what it is supposed to do for a patient). By replacing the hyponym-relation with type or role relations the erroneous substructure will be corrected.

Fellbaum has been manually examining all the cases of "Heart-shaped substructures" and found that many are in fact Type/Role distinctions.

4 Discussion and Conclusion

In this paper the most common hidden substructures in wordnet hierachical structure have been studied. A short overview of detected erroneous substructures found by other authors has been given. Authors offered a new substructure called "Heart-shaped substructure" that is pointing to possible errors in WordNet hierarchical structure. The pattern in Figure 5 has been evaluated by the experienced Fellbaum from the Department of Computer Scinece at Princeton University and she confirmed that it assists to identify errors in WordNet structures. In chapter 3 we presented a case study of heart-shaped substructure by Fellbaum.

Authors have also studied other error-pointing substructures in WordNets. More detailed survey of those did not fit into the frame of this short article, but we just mention some error-pointing substructures:

Separated trees with one to five levels which probably should be connected to other bigger tree. F.e. counting only one and two levels hyponym trees more the 200 ones have been found in both PrWN and EstWN.

The lowest level synset with many parents. This situation is necessarily not to be considered as a defect, but it can be when those synsets are expected to have more precise definitions, so they have only one parent.

Large closed subsets are the biggest separated pieces (connected componets) in bipartite graphs. In case of noun, hyponym realtion and PrWN the largest closed subset is 1,333 x 167. In case of EstWN with same POS and relation the largest closed subset is 4,945 x 457. The number 1,333 is as hyponyms in bipartite lower level and 167 is as hypernym in upper level. Special study these of such big chunks has been presented in another papers [8] [**?**].

Acknowledgements Acknowledgments goes here.

References

1. Brideaux, R.: Snappy Words, version 1.02, 04/06/2013. `http://www.snappywords.com///`
2. Fiser, D., Novak, J.: Visualizing slownet. In Proceedings: Electronic lexicography in the 21st century: eLex2011 pp. 76–82 (2011)
3. Hicks, A., Herol, A.: Cross-lingual evaluation of ontologies with rudify. Springer Berlin Heidelberg **128**, 151–163 (2011)
4. Horak, A., Pala, K., Rambousek, A., Povolny, M.: Debvisdic - first version of new client-server wordnet browsing and editing tool. In Proceedings of the Third International WordNet Conference - GWC 2006 pp. 325–328 (2006)
5. Levary, D., Eckmann, J.P., Moses, E., Tlusty, T.: Self reference in word definitions. CoRR **abs/1103.2325** (2011)

6. Liu, Y., Jiangsheng, Y., Zhengshan, W., Y., S.: Two kinds of hypernymy faults in wordnet: the cases of ring and isolator. Proceedings of the Second Global WordNet Conference pp. 347–351 (2004)
7. LogicalOctupus: Visuwords, 04/06/2013. http://www.visuwords.com//
8. Lohk, A., Tilk, O., Vohandu, L.: How to create order in large closed subsets of wordnet-type disctionaries. Estonian Papers in Applied Linguistics 9 pp. 149–160 (2013)
9. Louw, M.: Polaris User's Guide. Technical report (1998)
10. Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Introduction to wordnet: An on-line lexical database. International Journal of Lexicography **3**, 235–312 (1990)
11. Orav, H., Kerner, K., Parm, S.: Snapshot of estonian wordnet (in estionian). Keel ja Kirjandus **2**, 96–106 (2011)
12. Piasecki, M., Marcinczuk, M., Musial, A., Ramocki, R., Maziarz, M.: Wordnetloom: a graph-based visual wordnet development framework. In Proceedings of IMCSIT pp. 469–476 (2010)
13. Richens, T.: Anomalies in the wordnet verb hierarchy. Proceedings of the 22nd International Conference on Computational Linguistics: COLING-ACL 2008 pp. 729–736 (2008)
14. Smrž, P.: Quality control for wordnet development. Proceedings of the Second Global WordNet Conference pp. 206–212 (2004)
15. Thinkmap, Inc.: Visual Thesaurus, 04/06/2013. http://www.visualthesaurus.com/
16. Vider, K.: Estonian thesaurus - theory and reality (in estonian). Sõna tänapeva maailmas pp. 134–156 (2001)

APPENDIX E

Lohk, A.; Orav, H.; Võhandu, L. (2014). Some structural tests for WordNet with results. H. Orav, C. Fellbaum, P. Vossen (Edit.). Proceedings of the Seventh Global Wordnet Conference (313 - 317). Tartu University Press

Some Structural Tests for Wordnets, with Results

Ahti Lohk
Tallinn University of Technology
Akadeemia tee 15a
Tallinn, Estonia
ahti.lohk@ttu.ee

Heili Orav
University of Tartu
Liivi 2
Tartu, Estonia
heili.orav@ut.ee

Leo Võhandu
Tallinn University of Technology
Akadeemia tee 15a
Tallinn, Estonia
leo.vohandu@ttu.ee

Abstract

This paper proposes some test-patterns (viewed as sub-structures) to evaluate the hierarchical structure of wordnets. By observing hierarchical structure, both top-down and bottom-up experiments are carried out on four wordnets: Princeton WordNet (version 3.1), Cornetto (version 2.0), the Polish Wordnet (version 2.0) and the Estonian Wordnet (version 67). The top-down approach is used to find small hierarchies, which are defined as having up to three levels of subordinates starting from unique beginners (rootsynsets). The bottom-up perspective is looking at the links that appear due to polysemy, and yet these are not. These redundant links form "asymmetric ring topology", and should be eliminated. Finally, an additional particular feature of large closed subsets will be introduced. Addressed views provide an opportunity to evaluate and/or improve the structure of wordnet hierarchies. This paper also provides an overview of the current status of these four wordnets from the according to our proposed test patterns.

1 Introduction

No linguist doubts the importance of wordnets. There are currently about 60 different wordnets worldwide. There are different views on the amount of information that is put into the system of synsets. But Miller and Fellbaum's primary goal, to create a large hypernym/hyponym relational style synset system is the same everywhere. Groups of specialists are involved in every implementation of wordnet for a given language. Every specialist has her/his subjective view about the relational connections between synsets.

It is important that every team has a strong belief in the high quality of the system they have created.

The theory and practice of building and checking computer chips with many millions of elements has proven that one has to build an independent test system to check designer created connections. As wordnets are similarly complex systems, we aim to build such a test system for wordnets.

The task of tests is to create lists of different types of inconsistencies which any Wordnet has at the given moment. Structural inconsistencies do not always translate to a wordnet error. The last word in checking wordnet lists always belongs to a lexicographer. What is truly crucial is that such lists are comprehensive. Tests must check all structurally weak areas of a given wordnet at any given moment.

After a lexicographer has made needed corrections, there follows a repetition of the same test. Such an iterative process has only one goal – to come to a clear understanding of all the weak places a given test can find.

Every created test has a different power. Some tests point with 100% probability to an error made by a lexicographer, although the error rate is usually below 100%. Such tests also have an important lexicographic value, as a long list of inconsistencies usually points to a complicated linguistic problem lacking a unique solution.

In this article we study only hypernym/hyponym relations.

2 Background of the wordnets

2.1 Princeton WordNet (PrWN)

Wordnets (Fellbaum, 1998) have emerged as one of the basic standard lexical resources in the language technology field. Princeton WordNet (PrWN) and most other wordnets are structured into synsets. A synset is usually described as capturing a lexicalised concept. Synsets are linked by conceptual relations with names borrowed from linguistic work on lexical semantics, such as hypernymy, holonymy, meronymy and so on.

More than 60 languages followed suit for building wordnets for their vernacular and very different compilation strategies have been applied. Some teams have decided to translate PrWN and adjust the result of that translation. Some word-

net developers have chosen an opposite route, such as expanding from the most frequent words or from top concepts as it has seen in ontological approaches.

The following is a brief introductory description of three databases from the Fenno-Ugric language family, and the Germanic and Slavic branches of the Indo-European language family.

2.2 Cornetto

The goal of Cornetto[1] was to build a lexical semantic database for Dutch, following the structure and content of Wordnet and FrameNet. Cornetto comprises information from two electronic dictionaries: the *Referentie Bestand Nederlands*, which contains FrameNet-like structures, and the *Dutch wordnet* (DWN) which utilises typical wordnet structures. DWN has a similar structure as the English WordNet although the top-level hierarchy was developed from an ontological framework and more horizontal relations are defined. The database has 70,371 synsets and 119,108 lexical units.

2.3 Polish Wordnet (plWN)

Work on PolNet began in 2005 (Derwojedowa, 2008), and its thesaurus is currently composed of nearly 116,000 synonym sets. The plWN development was organised in an incremental way, starting with general and frequently used vocabulary. The most frequent words from a reference corpus of the Polish language were selected.

2.4 Estonian Wordnet (EstWN)

The Estonian Wordnet began as part of the EuroWordNet project (Vossen, 1998), and was built by translating base concepts from English to allow monolingual extension. Words (literals) to be included were selected on frequency basis from corpora. Extensions have been compiled manually from Estonian monolingual dictionaries and other monolingual resources. After the start several methods have been used, for example domain-specific, i.e there have been dealt with semantic fields like architecture, transportation etc, there are some endeavors to add derivatives automatically and the results have been used of sense disambiguation process. Version 67 of EstWN consists of 60,434 synsets, including 82,515 words.

[1] http://www2.let.vu.nl/oz/cltl/cornetto/index.html

3 Related works

The most similar research to our paper has been done by Tom Richens, who has studied the anomalies in the WordNet verb hierarchies (Richens, 2008). Under the notion of topological anomalies, he notes three types of sub-structures in the hierarachical structure of WordNet that should be checked: “cycles”, “rings” (these in turn are classified into “asymmetric ring topology” and “symmetric ring topology”) and “dual inheritance”. He emphasizes that if “dual ineritance” (which also includes “asymmetric ring topology” and “symmetric ring topology”) appears, it merits investigation.

In his paper, Richens refers to the work of Pavel Smrž (Smrž, 2004) and Yang Liu (Liu, 2004). Smrž proposes twenty-seven tests for quality control in wordnet development. In most cases these tests are dealing with editing errors like “empty ID, POS, SYNONYM, SENSE (XML validation)” or “duplicate literals in one synset”, but some of them are errors of hierarchical structure, like “cycles”, “dangling uplinks”, “structural difference from PWN and other wordnets”, “multi-parent relations”.

Lin proves and refers to two kind of hypernymy faults in WordNet (about version 2.0): rings and isolators, and asserts that “In the future, some amendments should be made to solve these issues during the evolution of WordNet” (Liu, 2004).

Research about quality and evaluation of WordNet are made also by Aron N. Kaplan et al. (Kaplan, 2001), Philippe Martin (Martin, 2003), Raghuvar Nadig (Nadig, 2008) and Tomáš Čapek (Čapek, 2012).

4 Top-down view, small hierarchies

A top-down view of the structure will begin walking through the unique beginner separating all hierarchical structures (see Fig. 2), which end after the root of the concept on three next levels. This view can be useful for detecting small hierarchies that have somehow remained unconnected to a higher hierarchy. A large number of small hierarchies points to a lack of feedback (see Table 1).

PrWN was originally constructed with 25 unique beginners (rootsynset). These rootsynsets were later connected to a single unique beginner labeled ”entity” (Miller, 2007). From Table 1, it can be seen that in the PrWN there are only 11

Princeton WordNet	
rootsynset	**352** (n-12, v-340, a-0)
1 add. level	155 (n-11, v-144, a-0)
2 add. levels	81 (n-0, v-81, a-0)
3 add. levels	48 (n-0, v-48, a-0)
Cornetto	
rootsynset	**497** (a-454, n-2, v-2, r-12, c-27)
1 add. level	285 (a-263, r-11, c-1)
2 add. levels	148 (a-137, r-1, c-10)
3 add. levels	40 (a-37, n-1, c-2)
Polish WordNet	
rootsynset	**861** (n-531, v-35, j-295)
1 add. level	586 (n-335, v-25, j-226)
2 add. levels	159 (n-100, v-9, j-50)
3 add. levels	49 (n-34, v-0, j-15)
Estonian WordNet	
rootsynset	**169** (n-129, v-4, a-36)
1 add. level	128 (n-94, v-0, a-34)
2 add. levels	18 (n-16, v-0, a-2)
3 add. levels	6 (n-6, v-0, a-0)

Table 1: Number of rootsynsets and number of hierarchies that have only up to three additional levels of subordinates. (Numbers in brackets are about parts of speech as it is shown in every WordNet database.)

noun root synsets with one additional level of hierarchy, which is probably either due to human error, or unfinished work.

According to Table 1, Cornetto has only two noun and two verb hierarchies. That shows that every added synset is located directly into a large hierarchy. (Rootsynsets for the nouns are iets:2 and niets:1, translated as ”something” and ”nothing”.)

The much smaller number of Estonian Wordnet’s rootsynsets (169) is due to the fact that the team has gradually started to take into account the specific nature of the information obtained by structural tests. For example, in version 65, the number of rootsynsets was 303. Most of the decrease in rootsynsets is due to the fall of noun rootsysets has been reduced from 248 to 129.

It may be wise to take advantage of the low number of verb root concepts of EstWN to improve other wordnets’ verb hierarchies. This is certainly the case when the number of root concepts is too big.

The number of small hierarchies can be reduced considerably trying to locate them in the bigger hierarchy. This approach is a particular issue in the noun and verb trees.

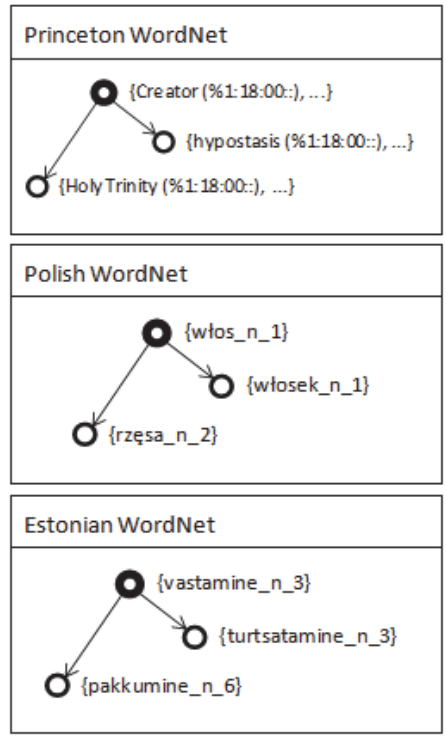


Figure 1: Small hierarchies. Rootsynsets with one additional level.

5 Bottom-up view, asymmetric ring topology

In this view, we are moving from lower level synsets to higher ones starting from synsets with many parents and separating substructures where such synsets are related to other synset directly and indirectly (see Fig. 2). The resulting subset is also referred to as a asymmetric ring topology (Richens, 2008) (see Table 2). This sub-structure may occur if lexicographers have created a new, more precise link to another synset, forgot to remove the previous relation. In this case one synset is connected to hypernym-synsets twice - directly and indirectly through other hypernym-synset (see Fig. 2)

6 The Largest Closed Subset (LGS)

LGS in hierarchical structures has been regarded as a coherent bipartite graph (Lohk, 2013).

	Synsets with many parents	Asymmetric ring topology
PrWN	1,425	30
Cornetto	2,438	306
plWN	10,942	476
EstWN	1,167	69

Table 2: Synsets with many parents and asymmetric ring topology numerically

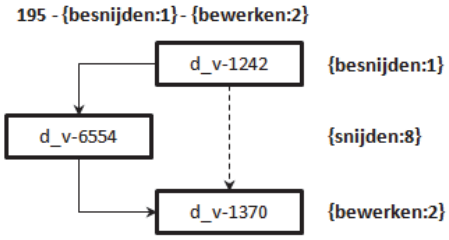


Figure 2: Asymmetric ring topology seen in Cornetto

In many cases LGS seems to be like particular feature of the hierarchical structure that links different hierarchical structures started from unique beginners. It is remarkable that in many cases the upper base of the bipartite graph consists of rootsynsets (see Table 3). Authors think that this conflict arises because the concepts of the root level are put to the same level with non-roots.

In Figure 3 an artificially constructed hierachical structure with one unique beginner (root node) has been shown. Closed subsets are highlighted by rectangles. Our interest is to find only the biggest ones, this is possible when a closed subsynset has at least two parents (represented with thick lines).

According to Figure 3 and Table 3 lower nodes in a closed subset are related to the first number in the second column of the table and upper nodes in a closed subset are related to the second number also in the second column of the table.

In the case of PrWN, every upper base synset in the bipartite graph belongs to the synset ”entity;” in the case of Cornetto, to ”iets:2” (in eng: ”something”); and in the case of EstWN into ”olev” (in eng: essive). Cornetto has one more large closed subset, related to verbs. As can be see in Table 1, the overall number of verb hi-

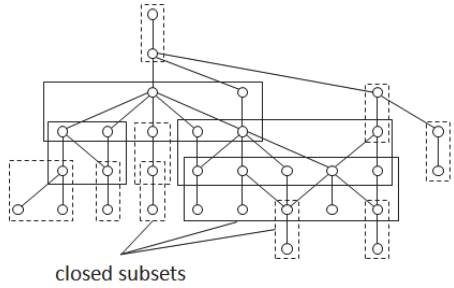


Figure 3: Artifically constructed tree of the WordNet with closed subsets

	The biggest closed subset	Root synsets in closed subset	Synsets of closed subsets that are connected to root synsets
PrWN	1,064 x 126	0	1
Cornetto[1]	11,032 x 589	0	1
Cornetto[2]	4,423 x 545	1	2
plWN	30,794 x 4,683	142	76
EstWN	1,526 x 66	8	1

Table 3: The largest closed subsets

erarchy is two and second big closed subset of Cornetto (in Table 3) connects these two (root synsets {afspelen:1, gebeuren:1, ..} and {zijn:7, uitmaken:2, vormen:5}).

While PrWN is obviously the most studied (see WordNet bibliography[2]) and Cornetto has a commercial version[3], it can be assumed that their hierarchical structure has received more attention (see Table 3, the number of rootsynsets in closed subset is in the case of PrWN and Cornetto 0).

Earlier tests with the Slovenian Wordnet (version 3.0) showed that a very large closed set may not be typical for all wordnets. It turned out that the largest closed subset size in this case was only 248 x 3.

LGS and closed subsets with many hypeonyms may be generally useful if the hypernyms in the upper base of closed sets are separated and their levels of concept are evaluated. Additionally, LGS seems to indicate the correctness (or uncorrectness) of the hierarchical structure, although this

[2] http://lit.csci.unt.edu/~wordnet/

[3] http://tst-centrale.org/nl/producten/lexica/cornetto/7-56

claim has not been definitively verified.

7 Discussion and Conclusion

The most difficult issue for wordnet compilers with regard to noun hierarchical relationships is to find the top hypernyms. The same also occurs in regard to finding the top concepts for the most frequent verbs, both transitive and intransitive. As for adjectives, the situation is even more unclear, as wordnets for various languages deal with adjectives differently. In some wordnets, adjectives are hierarchical (as seen in Table 1: Cornetto, EstWN), but in PWN, adjectives have different types of" semantic connections.

One analyses only the short hierarchies in all wordnet variants, (root level plus up to 3 lower levels) one comes to the realisation that new add-ons for wordnets have created a situation in which missing feedback has lost the information required to correctly connect synsets.

All wordnets studied here show that the expansion process requires strong and effective feedback.

As is made clear by Table 1, in the top-down perspective, three of the four wordnets studied here require either verb or noun hierarchy correction. However, as Cornetto has only two hierarchies for nouns and verbs, it has somehow excluded small hierarchies. This shows that Cornettos team is using different tools or/and ways for add-ons.

References

Magdalena Derwojedowa, Maciej Piasecki, Stanislaw Szpakowicz, Magdalena Zawislawska and Bartosz Broda. 2008. *Words, Concepts and Relations in the Construction of Polish WordNet* In Proceedings of the Fourth Global WordNet Conference - GWC 2008. pp: 162–177.

Tomáš Čapek. 2012. *SENEQA – System for Quality Testing of Wordnet Data.* Proceedings of 6th International Global Wordnet Conference. Matsue, Japan, 9-13 January 2012. pp: 400-404.

Christiane D. Fellbaum. 1998. *WordNet An Electronic Lexical Database* Cambridge, Massachusetts, London, England: The MIT Press

Aaron N. Kaplan and Lenhart K. Schubert. 2001. *Measuring and improving the quality of world knowledge extracted from WordNet.* University of Rochester, Rochester, NY.

Yang Liu, Jiangsheng Yu, Zhengshan Wen and Shiwen Yu. 2004. *Two Kinds of Hypernymy Faults in WordNet: the Cases of Ring and Isolator.* Proceedings of the Second Global WordNet Conference. Brno, Czech Republic, 20-23 January 2004. pp: 347-351.

Ahti Lohk, Ottokar Tilk and Leo Võhandu . 2013. *How to Create Order in Large Closed Subsets of WordNet-type Dictionaries* Estonian Papers in Applied Linguistics 9 pp: 149–160.

Philippe Martin. 2003. *Correction and extension of WordNet 1.7* Conceptual Structures for Knowledge Creation and Communication: Springer. pp: 160–173.

George A. Miller. and Christiane D. Fellbaum. 2007. *WordNet then and now* Lang Resources & Evaluation, Volume 41, Issue 2. pp: 209–214.

Nadig Raghuvar, Ramanand J and Bhattacharyya Pushpak. 2008. *Automatic Evaluation of Wordnet Synonyms and Hypernyms* Proceedings of ICON-2008: 6th International Conference of Natural Language Processing.

Tom Richens. 2008. *Anomalies in the wordnet verb hierarchy* Proceedings of the 22nd Inter-national Conference on Computational Linguistics: COLING-ACL 2008. pp: 729–736.

Piek Vossen. 1998. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks* Dordrecht: Kluwer Academic Publishers.

Pavel Smrž. 2004. *Quality Control for Wordnet Development.* Proceedings of the Second Global WordNet Conference. Brno, Czech Republic, 20-23 January 2004. pp: 206-212.

APPENDIX F

Lohk, A; Norta, A; Orav, H; Võhandu, L. (2014). New Test Patterns to Check the Hierarchical Structure of Wordnets. G. Dregvaite, R. Damasevicius (Edit.). Information and Software Technologies: 20th International Conference, ICIST 2014, Druskininkai, Lithuania, October 9-10, 2014. Proceedings (110 - 120). Springer

New Test Patterns to Check the Hierarchical Structure of Wordnets

Ahti Lohk[1], Alexander Norta[1], Heili Orav[2], and Leo Võhandu[1]

Tallinn University of Technology, Akadeemia tee 15a, 12618 Tallinn, Estonia[1]
University of Tartu, J. Liivi 2, 50409, Tartu, Estonia[2]
{ahti.lohk,alexander.norta,leo.vohandu}@ttu.ee
heili.orav@ut.ee

Abstract. The goal of this paper is to introduce test patterns for checking inconsistencies in the hierarchical structure of wordnets. Every test pattern (displayed as a substructure) points out the cases of multiple inheritance and two of them are studied in depth by expert linguists, or lexicographers. Furthermore, this research associates test patterns with the inconsistencies they help to detect in wordnets, and presents instances of the test patterns. All examples use the Estonian Wordnet (Versions 66 or 67), some results we are shown for the Princeton WordNet (Version 3.1).

Keywords: wordnet, hierarchical structure, evaluation, test patterns, multiple inheritance

1 Introduction

Many tasks of natural language processing, such as machine translation, information retrieval and word sense disambiguation use wordnets as a lexical resource. Therefore, wordnets are attractive due to their hierarchical structure of lexical concepts. Unfortunately, there are no good methods to study the condition of its hierarchical structure. Richens [1] and Liu [2] describe two different types of rings in the substructure of the wordnet hierarchy that point to inconsistencies like a wrongly inherited domain category or ignoring the principle of economy. A common denominator of these two type of rings is that they consist of multiple inheritance cases.

With respect to the state of the art, research has been conducted for individually testing the hierarchy substructures of wordnets. For example, David Levary gives an overview of the loops and self-references in the hierarchical structure of wordnets [3]. Liu [2] and Richens [1] show all rings of asymmetric and symmetric nature in a ring topology that is based on the same structure. In Smrž [4], the author presents 27 tests for quality control in wordnet development. Only some of those tests are for checking errors in the hierarchical structure, like ”cycles”, ”dangling uplinks”, ”structural difference from the Princeton WordNet and other wordnets”, or ”multi-parent relations”. However, there are no test pattern systems that would help to investigate a hierarchical structure in a general way, especially in the case of multiple inheritance.

adfa, p. 1, 2011.

This paper fills the gap in the state-of-the-art by asking the main research question of how test patterns help to check and evaluate the multiple inheritance in the hierarchical structure of wordnets. In order to answer the question, we present different test patterns as different views on the substructures of the wordnet hierarchy in the cases of multiple inheritance. The need to check the hierarchical structure emerges because of wordnet extensions with new concepts and semantic relations that either happen manually [5], semi-automatically [6, 7], or fully automatically [8, 9]. Thus, every pattern reveals different inconsistencies in the hierarchical structure. The majority of inconsistency cases are caused by redundant, missing or wrong semantic relations between synsets. The utility of the patterns lies in supporting expert linguists who check substructures after the extensions in any human language wordnet.

We structure the paper as follows: Section 2 gives additional background for understanding the main body of the paper. Next, Section 3 shows test patterns for checking the wordnets. Section 4 discusses the inconsistency taxonomy related to these test patterns. Section 5 evaluates the test patterns providing a numerical overview and finally, Section 6 concludes the paper and presents future work.

2 Features of Wordnet-like Dictionaries

Wordnets share properties for the concepts of polysemy that are a part of the definitions of the test patterns. On the other hand, regular polysemy is only a part of one test pattern definition, namely of the pattern *dense component*. In the remainder, Section 2.1 gives general structural features for wordnet and Section 2.2 polysemy versus regular polysemy.

2.1 Wordnet-like dictionaries

The fundamental approach for designing WordNet-like dictionaries came from the Princeton WordNet [10]. Each wordnet shares certain structural features. First, synonym sets (synsets) group many synonyms that share the same meaning and are referred to as concepts. Semantic relations connect synsets to each other, e.g. by hypernymy, meronymy for creating a hierarchical structure, and caused by, near synonym that do not create a hierarchical structure. In this article, we consider only hypernymy-hyponymy relations as the objects of analysis. Furthermore, there is no extension limitation for approaching different semantic relations that shape the hierarchical structure.

For details about Estonian Wordnet, we refer the reader to [5]. Furthermore, Princeton WordNet has 117,773 synsets and 88,721 hypernym-hyponym relations. In Estonian Wordnet Version 66, these values are 58,566 and 51,497 respectively, while for Versions 67, the values are 60,434 and 52,678, respectively. Princeton WordNet has hypernym-hyponym relations only in the cases of nouns and verbs; in the Estonian Wordnet in the case of nouns, verbs and adjectives.

2.2 Regular polysemy vs the regularity of multiple inheritance

According to Ravin and Leacock [11], polysemy is the multiplicity of meanings of words. The best-known definition of regular (also systematic or logic) polysemy is given by Apresjan [12]. According to Langemets [13], regular polysemy is a status where at least two words have at least two meanings with a similar relation between those meanings. For example, if the word school has the meaning institution and building, then the same is true about a hospital. The latter is also an institution as well as a building. According to Freihat et al. [14], institution building is an example of a polysemic pattern.

Multiple inheritance in wordnet hierarchies is the case where one synset has at least two parents, i.e., the synset inherits properties from many concepts. The regularity of multiple inheritance is comparable to regular polysemy in that instead of words, there are synsets and instead of a polysemic pattern, there exists a pattern of many parents. It is important to mention that if synsets are singletons, then there is no difference between the meanings of regular polysemy and the regularity of multiple inheritance.

Next, the set of test patterns for checking wordnet hierarchy-inconsistency is given.

3 Test Patterns

For every form of inconsistency, we will give a specific test pattern that is applicable to every language wordnet. Every pattern addresses a specific substructure of the hierarchical structure in wordnets and has the property of multiple inheritance, i.e. polysemy. For the sake of test pattern set's completeness, the patterns presented in Section 3.1 are inspired by Liu and Richens [2, 1] while the remainder are entirely original work.

3.1 Rings

This pattern is a substructure where one superordinate has a subordinate via two branches, e.g. in Figure 1 and 2, U_1 has the subordinate L_1. We distinguish two types of rings. In the case of a symmetric ring topology (SRT) the lengths of all chains in the branches are equal, i.e. m = n in Figure 1. In an asymmetric ring topology (ART) the lengths are different, i.e. m ≠ n in Figure 2. Note that while Figures 1 and 2 only show two branches, this pattern extends to more branches.

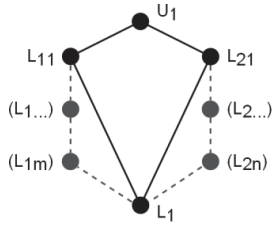


Fig. 1. Pattern of a symmetric ring topology

When a synset has information about a domain category, then both types of rings allow to detect a certain inconsistency automatically, e.g. in a situation where L_{1m} and L_{2n} are from different domain categories. Research in [2] confirms that one synset as a concept cannot inherit properties from same domain categories.

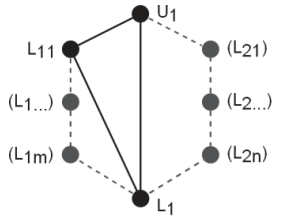


Fig. 2. Pattern of an asymmetric ring topology

The asymmetric ring topology with one redundant branch as in the center of Figure 2 indicates that the branch connecting U_1 to L_1 is not allowed because this connections already transitively exists via L_{11} [10].

3.2 Closed subset (CS)

A modified equivalence-class-finding algorithm [15] yields the following pattern. As Figure 3 shows, and based on the sequence of hypernym relations, our algorithm separates all coherent bipartite graphs. The inconsistency occurs when the location information about the root synset equals the upper level of a bipartite graph, e.g. U_1 in Figure 3. This information indicates that the upper base involves concepts that should be on different levels. Thus, a root synset may either be added to a higher level, or connected to pre-existing higher-level concepts.

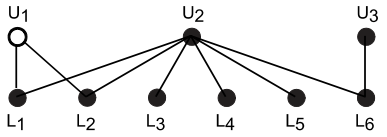


Fig. 3. Pattern of a closed set

3.3 Dense component (DC)

A dense component is a substructure of the hierarchical structure of wordnets that has at least two synsets with at least two identical parents. Every such kind of substructure presents the case of regular polysemy, i.e., systematic polysemy. Therefore, in the evaluation process, expert linguists/lexicographers have to check if regular polysemy is justified or not. The lower level synsets in Figure 4, L_1 and L_3, have at least two identical parents, U_2 and U_3. Additionally, dense components may have synsets in common that

have at least two parents in the upper level's set of nodes. For example in Figure 4, L_1 and L_3 have in common not only the synset L_2 but also the nodes U_1 to U_4 from the upper level. Separating this information keeps the polysemic context clear while every dense component is presented with related synsets to simplify the work of expert linguists/lexicographers.

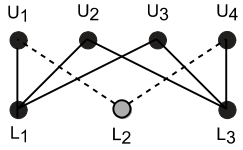


Fig. 4. Pattern of a dense component

3.4 Heart-shaped substructure (HSS)

In the case of a heart-shaped substructure, two upper level synsets have one common subordinate directly in common. For example in Figure 5, U_1 and U_2 have one common L_1 and simultaneously the upper level synsets also have L_3 partially transitively in common via L_2. Linguists from Princeton University have found that this pattern is helpful for detecting wrong semantic relations, mostly role and type relations. Unfortunately, a complete analysis has not been done for the Estonian Wordnet yet, but Figure 11 shows an example.

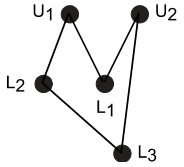


Fig. 5. Pattern of a heart-shaped substructure

3.5 Substructures through a synset member or a part of a compound word (COM)

The Estonian Wordnet consists of many cases where an upper level synset's lexical unit relates to the main word in a compound word which is a member of the subordinates set, e.g. U_2 to L_1 to L_5 in Figure 6. Additionally, an upper level synset's lexical unit may also relate to subordinates that have the same lexical unit, e.g. L_1 to L_5. Furthermore, this pattern must simultaneously have at least one additional superordinate, e.g. U_1 from among L_1 to L_5 in Figure 6.

To evaluate this kind of pattern, expert linguists/lexicographers must ask: if U_1 is connected to L_1, why it is not connected to L_2, L_3, L_4 or L_5? This question helps to make a decision regarding inconsistencies that this pattern may have. In case of the Estonian Wordnet, we found that sometimes this pattern points to a situation where a subordinate should have additional synsets with different meanings, e.g. L_1.

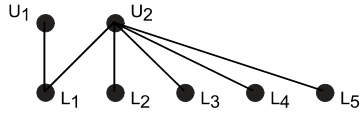


Fig. 6. Pattern of COM

In the next section, we will introduce the inconsistencies that relate to respective test patterns.

4 Inconsistencies of Substructures

The purpose of our test patterns is to check after presenting a pattern if an instance of that wordnet's substructure contained inconsistencies. Thus, in Section 4.1, we will list these inconsistencies and in Section 4.2, we will give examples.

4.1 Inconsistency taxonomy

Any correction a linguist/lexicographer carries out affects a substructure by either deleting, inserting, merging or modifying a synset.

1. **Regularity of polysemy** – in accordance with Section 2.2, linguists have to check if the regularity of multiple inheritance is justified or not. Linguists also check which synsets have to be connected to a pattern of parents.
2. **Ignoring the principle of economy (redundant semantic relation)** – for building a wordnet as a lexical inheritance system, we consider the following: every synset in a wordnet hierarchy has to be connected to the nearest concept. Here we focus on cases where one synset (S1) is connected to specific parents (S2) and at the same time to parents that are ancestors (S3) to both synsets (S1 and S2).
3. **Inappropriate semantic relationship** – it implies that the semantic relationship's type must change. Atserias et al. [16] point to a situation that occurs in Princeton-WordNet ”the IS-A link is used to code other types of relations (e.g. similar or place)”. The same problem holds for a role and type relation that wordnets have not defined yet [17].
4. **Wrongly inherited domain category** – if one synset inherits many different concepts from different domain categories, at least one of them represents an exception to the linguistic theory [2] that a concept has to inherit properties only from the super-concept of the same domain category. The gloss of the synset indicates which of the categories is most appropriate [10].
5. **Root synset on the wrong level** – this is a sub-problem of the unique-beginners problem that Smrž [4] defines and means that dangling uplinks occurred.

The assignment options of inconsistencies to test patterns are given in Table 1 comprises. The sequence numbers in the first column correspond to the inconsistency enumeration above while the test pattern abbreviations are given in the first row.

Table 1. The kinds of inconsistencies the test patterns help to detect

		ART	SRT	CS	DC	HSS	COM
1	Regularity of multiple inheritance				x+		
2	Ignoring the principle of economy	x+			x		
3	Inappropriate semantic relation	x	x	x	x	x+	x+
4	Wrongly inherited domain category	x+	x+	x	x	x	x
5	Root synset on the wrong level			x+			

The symbol ”+” added to the table cells in addition to *x* denotes that a respective test pattern is particularly suitable for detecting a specific inconsistency type. Note that column ART has two ”+” assignments as Figure 2 shows two different examples, namely with and without a redundant link.

4.2 Some examples

In this subsection, we will present the examples that cover the test patterns that are given in Section 3. Figure 7 represents the case of an asymmetric ring topology with an empty branch. Here the *human* is connected to *bootlegger* directly (dotted line) and indirectly.

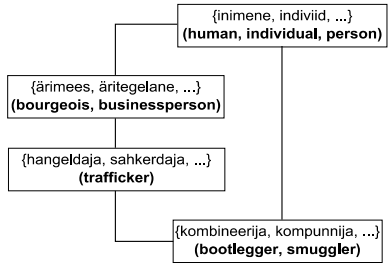


Fig. 7. Instance of an asymmetric ring topology

According to the understanding that a wordnet is a lexical inheritance system, only the nearest concepts in the hierarchy have to be connected. Therefore, in Figure 7, the dotted line as a connection between the specific *bootlegger* and the too general *human* is redundant.

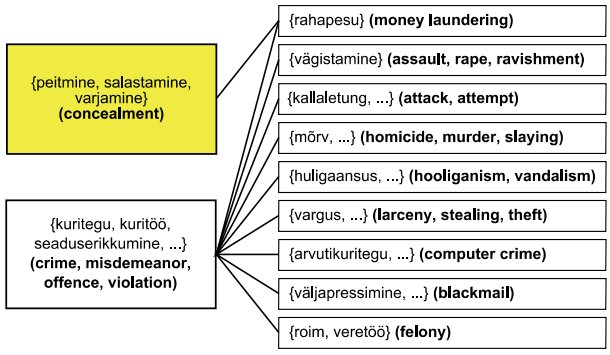


Fig. 8. Instance of a closed subset

Figure 8 depicts a closed subset. Two general concepts are related to specific ones. The concept with a colored background indicates to the root synset or unique-beginner case or concept without any parents. In order to solve this situation, this kind of dangling uplink needs to be connected to a more general concept.

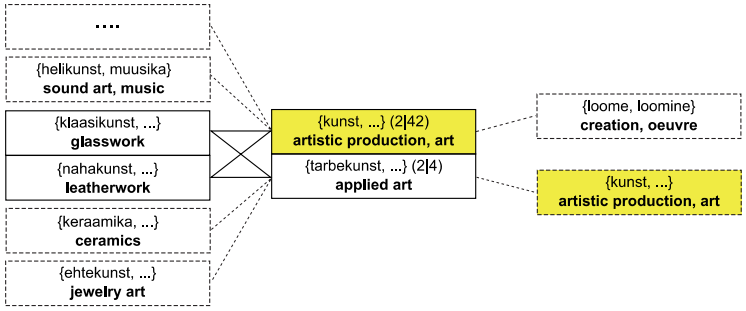


Fig. 9. Instance of a dense component

Figure 9 presents an example of a dense component where the dashed lines present background information and the colored background points to the same concept. This kind of additional information is for the linguist who does not need to check the wordnet management system for the background of every instance of a dense component.

As the co-hypernyms are concepts of a different level, due to *artistic production* involving also *applied art*, it means „kunst“ must be the parent of „tarbekunst“ and the links between „kunst“ and „klaasikunst“ and also „kunst“ and „nahakunst“ are redundant. In Figure 10, the key synset is „madu“ (*serpent*) that is included in three compound words as „boamadu“ (*boa*), „lõgismadu“ (*Crotalus*) and „mürkmadu“ (*Vipera aspis*). „Boamadu“ (*boa*), in turn, simultaneously has the superconcept of „sall“ (*scarf*).

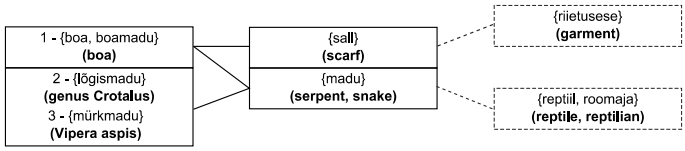


Fig. 10. A substructure via a synset member or a part of a compound word

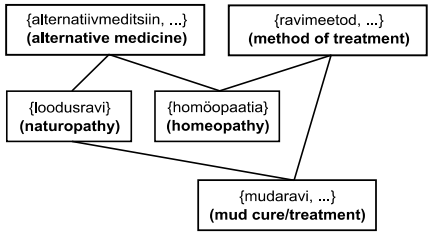


Fig. 11. Instance of a heart-shaped substructure

Finally, Figure 11 shows an instance of a heart-shaped substructure. The question arises why "homöopaatia" (*homeopathy*) is not a subcase of "loodusravi" (*naturopathy*). Secondly, are "mudaravi" (*mud cure*) and "homöopaatia" (*homeopathy*) subcases of "alternatiivmeditsiin" (*alternative medicine*) or of "ravimeetod" (*method of treatment*)? On the basis of the definitions of these concepts, lexicographers decided that both are subcases of the method of treatment and that *alternative medicine* is connected to them via a *holonymy* relation.

5 Evaluation

We focused on two test patterns, namely the dense component (DC) and asymmetric ring topology with index zero (ART_0), i.e. with a redundant link as depicted in Figure 2. Since the test patterns overlap, correcting the instance of the dense components test pattern also affects other test pattern instances, as shown in Table 2. The test pattern system of the Estonian Wordnet Version 66 indicated that the number of multiple inheritance cases reduced from 1,677 to 1,164 in comparison to the Estonian Wordnet Version 67.

Table 2. The number of occurrences of test patterns

		EstWN (v66)	EstWN (v67)	PrWN (v3.1)
1	ART_0	119	79	41
2	ARTx	821	611	1,181
3	SRT	567	270	531

4	CS	21	11	9
5	DC	121	24	107
6	HSS	450	167	149
7	COM	932	406	366

In the process of using all test patterns to check the wordnets the lexicographer has to use the following typical actions:

- add a new synset
- merge synsets
- remove a synset
- add or remove lexical units from a synset
- change a semantic relation
- add or remove a semantic relation

These actions usually take place through the wordnet management system and will be repeated after every extensive change in the hierarchical structure of wordnet.

6 Conclusion

In this paper, we proposed the use of test patterns for detecting inconsistencies in the substructures of wordnet hierarchies. After specifying how these patterns relate to the types of inconsistencies, examples of real cases demonstrated test pattern applications. In the evaluation, we showed that test pattern application yields many reductions in inconsistencies in the substructures of wordnet hierarchies. Consequently, linguists and lexicographers have a set of heuristics available for locating inconsistencies faster.

Different test patterns, covering often same hierarchical structures (but in different perspective) help to check wordnet hierarchy in the multiple inheritance cases. It turns out, those different perspectives point to different type of inconsistencies intended to evaluate for lexicographer. Lexicographer evaluates the instances of test patterns and if it is needed, corrects the wordnet hierarchical substructure, which the test patterns are pointing. The test pattern system introduced here helps to detect at least five different kinds of inconsistencies: *regularity of multiple inheritance, ignoring the principle of economy, inappropriate semantic relation, wrongly inherited domain category, root synset on the wrong level*. In order to solve these problems, the lexicographer has to typically add, remove, or change the semantic relations or synsets.

After the first correction of wordnet hierarchical structure through test patterns, the same process may repeat.

As future work, we plan to investigate wordnets further to come closer to pattern-set completeness. Additionally, the currently conceptually specified patterns must be formalized. That way it would be possible to meaningfully automate the detection of patterns in wordnet substructures, which would also include a recommendation system for inconsistency detection.

7 References

1. Richens, T., Anomalies in the WordNet verb hierarchy. In: Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1, pp. 729–736. Manchster, UK (2008)
2. Liu, Y., Yu, J., Wen, Z., Yu, S.: Two kinds of hypernymy faults in WordNet: the cases of ring and isolator. In: Proceedings of the Second Global WordNet Conference, pp. 347–351. Brno, Czech Republic (2004)
3. Levary, D., Eckmann, J.-P., Moses, E., Tlusty, T.: Loops and Self-Reference in the Construction of Dictionaries. Phys. Rev. X. Vol. 2(3), pp. 031018-1–031018-10 (2012)
4. Smrž, P.: Quality Control and Checking forWordnet Development: A Case Study of BalkaNet.Romanian Journal of Information Science and Technology, Vol. 7, pp. 1–9. Romania (2004)
5. Orav, H., Kerner, K., Parm, S.: Snapshot of Estonian Wordnet. Keel ja Kirjandus, pp. 96–106. Estonia (2011)
6. Navigli, R.: Semi-Automatic Extension of Large-Scale Linguistic Knowledge Bases. FLAIRS Conference, pp. 548–553. Florida, USA (2005)
7. Beneventano, D., Bergamaschi, S., Sorrentino, S., Extending WordNet with compound nouns for semi-automatic annotation in data integration systems. In: International Conference on Natural Language Processing and Knowledge Engineering, pp. 1–8, IEEE, Dalian, China (2009)
8. Sagot, B., Fišer, D.: Extending wordnets by learning from multiple resources. In: LTC'11: 5th Language and Technology Conference, pp. 526–530. Poznan, Poland (2011)
9. Mihàltz, M., Sass, B., Indig, B.: What Do We Drink? Automatically Extending Hungarian WordNet With Selectional Preference Relations. Joint Symposium on Semantic Processing, pp. 105–109 (2013)
10. Fellbaum, C., D.: WordNet: An Electronic Lexical Database. The MIT Press, Cambridge, Massachusetts, London, England (1998)
11. Ravin, Y., Leacock, C.: Polysemy: Theoretical and computational approaches. MIT Press (2000)
12. Apresjan, J., D.: Regular polysemy. Linguistics, vol. 12(142), 5–32 (1974)
13. Langemets, M.: Nimisõna süstemaatiline polüseemia eesti keeles ja selle esitus eesti keelevaras. Eesti Keele Sihtasutus. Tallinn, Estonia (2010)
14. Freihat, A. A., Giunchiglia, F., Dutta, B.: Approaching Regular Polysemy in WordNet. In: The Fifth International Conference on Information, Process, and Knowledge Management, pp. 63–69. Nice, France (2013)
15. Knuth, D., E.: The Art of Computer Programming. Volume 1, Addison-Wesley (2012)
16. Atserias, J., Climent, S., Mor´e, J., Rigau, G.: A proposal for a Shallow Ontologization of WordNet. In: Proceedings of the 21th Annual Meeting of the Sociedad Espa˜nola para el Procesamiento del Lenguaje Natural, Vol. 5, pp. 161–167, Granada, Spain (2005)
17. Lohk, A., Võhandu, L.: Independent Interactive Testing of Interactive Relational Systems. Man-Machine Interactions 3, pp. 63–70. Springer International Publishing (2014)

CURRICULUM VITAE

1. Personal Data

Name: Ahti Lohk
Date of birth: 31.03.1975
Place of birth: Pärnu, Estonia
Citizenship: Estonia
E-mail address: ahti.lohk@ttu.ee

2. Education

Educational institution	Graduation year	Education (field of study/degree)
Tallinn University of Technology	2008	M.Sc. in Informatics
Tallinn University of Technology	2003	B.Sc. in Civil Engineering

3. Language competence/skills (fluent, average, basic skills)

Language	Level
Estonian	Fluent
English	Average
Russian	Average

4. Special courses

Period	Educational or other organisation
July 30 – August 3, 2012	International Summer School in Language and Speech Technologies (SSLST); Tarragona, Spain
August 19 – 23, 2012	11th Estonian Summer School on Computer and Systems Science (ESSCaSS); Jäneda, Estonia
July 22 – 26, 2013	International Summer School on Trends in Computing (SSTiC); Tarragona, Spain

5. Professional employment

Period	Organisation	Position
2007 –	Institute of Estonian Language	Part-time programmer
2003 –	Tallinn University of Technology	Research and Teaching Assistant
1997 – 2003	Tallinn University of Technology	Part-time lecturer
2001 – 2011	Tallinn Technical Secondary School	Teacher of programming

6. Scientific work

Lohk, A.; Orav, H.; Võhandu, L. (2014). Some Structural Tests for WordNet with Results. H. Orav, C. Fellbaum, P. Vossen (Editors). Proceedings of the Seventh Global Wordnet Conference (313 - 317).Tartu University Press

Lohk, A.; Võhandu, L. (2014). Independent Interactive Testing of Interactive Relational Systems. A. Gruca, T. Czachórski, S. Kozielski (Editors). Man-Machine Interactions 3 (63 - 70).Springer

Lohk, A; Norta, A; Orav, H; Võhandu, L. (2014). New Test Patterns to Check the Hierarchical Structure of Wordnets. G. Dregvaite, R. Damasevicius (Editors). Information and Software Technologies: 20th International Conference, ICIST 2014, Druskininkai, Lithuania, October 9-10, 2014. Proceedings (110 - 120). Springer

Lohk, A.; Allik, K.; Orav, H.; Võhandu, L. (2014). Dense Components in the Structure of WordNet. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14): LREC2014, Reykjavik, Iceland, May 26-31, 2014. (Editors) Nicoletta Calzolari and Khalid Choukri and Thierry Declerck and Hrafn Loftsson and Bente Maegaard an. ELRA, 1135 - 1139.

Lohk, A.; Tilk, O.; Võhandu, L. (2013). How to Create Order in Large Closed Subsets of Wordnet-type Dictionaries. Eesti Rakenduslingvistika Ühingu aastaraamat, 9, 149 - 160.

Leivo, M.; Lohk, A.; Ross, K.; Tafenau, K. (2013). Johannes Gutslaffi piiblitõlge. Eesti Keele Sihtasutus

Lohk, A.; Võhandu, L. (2012). Eesti wordnet'i struktuuri analüüsist. Eesti Rakenduslingvistika Ühingu aastaraamat, 8, 139 - 151.

Lohk, A.; Vare, K.; Võhandu, L. (2012). Visual Study of Estonian Wordnet using Bipartite Graphs and Minimal Crossing Algorithm. In: Proceedings of 6th International Global Wordnet Conference: 6th International Global Wordnet Conference: Matsue, Japan, pp. 167 - 173.

Lohk, A.; Vare, K.; Võhandu, L. (2012). First Steps in Checking and Comparing Princeton WordNet and Estonian Wordnet. In: Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH: EACL 2012; April 23 - 24 2012; Avignon France, pp. 25 - 29.

Täks, E.; Võhandu, L.; Lohk, A.; Nyman-Metcalf, K.; Rull, A. (2012). A Tool for Exploring the Hidden Structure of Legislation. In: Proceedings of the fundamental concepts and the systematisation of law. Workshop at Jurix 2011 in Vienna. JURIX 2011: The 24th International Conference on Legal Knowledge and Information Systems. University of Vienna.

Täks, E.; Võhandu, L.; Lohk, A.; Liiv, I. (2011). An Experiment to Find a Deep Structure of Estonian Legislation. Katie M. Atkinson (Editors). Legal Knowledge and Information Systems - JURIX 2011: The Twenty-Fourth Annual Conference (93 - 102). IOS Press

Täks, E.; Lohk, A. (2010). An Alternative Method for Computerized Legal Text Restructuring. Radboud G. F. Winkels (Editors). Legal Knowledge and Information Systems - JURIX 2010: The Twenty-Third Annual Conference (171 - 174). IOS Press

Lohk, A.; Võhandu, L. (2010). On the Estonian Emotion Vocabulary as It Is Presented in the Dictionary of Synonyms. In: Book of Abstracts: International Workshop "Emotions in and Around Language", Tallinn, 23-24. Sept. 2010. Tallinn, pp. 50.

7. Defended theses

„Advanced Text Search Capabilities in the Example of the Bible: Determining Search Functionalities and Preparatory Works“
M.Sc. (Informatics), Tallinn University of Technology, 2008

„Using Object-oriented Modelling to Build Engineering Software“
B.Sc. (Civil Engineering), Tallinn University of Technology, 2003

8. Supervised theses

Kaur Laanemäe, “A method to analyse tonality of news feeds”, Master thesis, 2015

9. Other academic activities

2012 – ... Reviewer of the yearbook of Estonian Association for Applied Linguistics (EAAL)

ELULOOKIRJELDUS

1. Isikuandmed

Ees- ja perekonnanimi: Ahti Lohk
Sünniaeg: 31.03.1975
Sünnikoht: Pärnu, Eesti
Kodakondsus: Eesti
E-posti aadress: ahti.lohk@ttu.ee

2. Hariduskäik

Õppeaasta	Lõpetamise aasta	Haridus (eriala/kraad)
Tallinna Tehnikaülikool	2008	Tehnikateaduste magister (Informaatika)
Tallinna Tehnikaülikool	2003	Tehnikateaduste bakalaureus (Ehitustehnika)

3. Keelteoskus (alg-, kesk- või kõrgtase)

Keel	Tase
Eesti keel	Kõrgtase
Inglise keel	Kesktase
Vene keel	Kesktase

4. Täiendusõpe

Periood	Täiendusõppe korraldaja nimetus
30 juuli – 3 august, 2012	Rahvusvaheline suvekool, keele- ja kõnetehnoloogiad (SSLST); Tarragona, Hispaania
19 – 23 august, 2012	Eesti suvekool, Arvuti- ja süsteemiteadus (ESSCaSS); Jäneda, Eesti
22 – 26 juuli, 2013	Rahvusvaheline suvekool, Andmetöötluse trendid (SSTiC); Tarragona, Hispaania

5. Teenistuskäik

Periood	Tööandja nimetus	Ametikoht
2007 –	Eesti Keele Instituut	Osalise tööajaga programmeerija
2003 –	Tallinna Tehnikaülikool	Assistent
1997 – 2003	Tallinna Tehnikaülikool	Tunnitasuline õppejõud
2001 – 2011	Tallinna Tehnikagümnaasium	Programmeerimise õpetaja

6. Teadustegevus

Lohk, A.; Orav, H.; Võhandu, L. (2014). Some Structural Tests for WordNet with Results. H. Orav, C. Fellbaum, P. Vossen (Toim.). Proceedings of the Seventh Global Wordnet Conference (313 - 317).Tartu University Press

Lohk, A.; Võhandu, L. (2014). Independent Interactive Testing of Interactive Relational Systems. A. Gruca, T. Czachórski, S. Kozielski (Toim.). Man-Machine Interactions 3 (63 - 70).Springer

Lohk, A; Norta, A; Orav, H; Võhandu, L. (2014). New Test Patterns to Check the Hierarchical Structure of Wordnets. G. Dregvaite, R. Damasevicius (Toim.). Information and Software Technologies: 20th International Conference, ICIST 2014, Druskininkai, Lithuania, October 9-10, 2014. Proceedings (110 - 120). Springer

Lohk, A.; Allik, K.; Orav, H.; Võhandu, L. (2014). Dense Components in the Structure of WordNet. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14): LREC2014, Reykjavik, Iceland, May 26-31, 2014. (Toim.) Nicoletta Calzolari and Khalid Choukri and Thierry Declerck and Hrafn Loftsson and Bente Maegaard an. ELRA, 1135 - 1139.

Lohk, A.; Tilk, O.; Võhandu, L. (2013). How to Create Order in Large Closed Subsets of Wordnet-type Dictionaries. Eesti Rakenduslingvistika Ühingu aastaraamat, 9, 149 - 160.

Leivo, M.; Lohk, A.; Ross, K.; Tafenau, K. (2013). Johannes Gutslaffi piiblitõlge. Eesti Keele Sihtasutus

Lohk, A.; Võhandu, L. (2012). Eesti wordnet'i struktuuri analüüsist. Eesti Rakenduslingvistika Ühingu aastaraamat, 8, 139 - 151.

Lohk, A.; Vare, K.; Võhandu, L. (2012). Visual Study of Estonian Wordnet using Bipartite Graphs and Minimal Crossing Algorithm. In: Proceedings of 6th International Global Wordnet Conference: 6th International Global Wordnet Conference: Matsue, Japan, pp. 167 - 173.

Lohk, A.; Vare, K.; Võhandu, L. (2012). First steps in checking and comparing Princeton WordNet and Estonian Wordnet. In: Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH: EACL 2012; April 23 - 24 2012; Avignon France, pp. 25 - 29.

Täks, E.; Võhandu, L.; Lohk, A.; Nyman-Metcalf, K.; Rull, A. (2012). A tool for exploring the hidden structure of legislation. In: Proceedings of the fundamental concepts and the systematisation of law. Workshop at Jurix 2011 in Vienna. JURIX 2011: The 24th International Conference on Legal Knowledge and Information Systems. University of Vienna.

Täks, E.; Võhandu, L.; Lohk, A.; Liiv, I. (2011). An Experiment to Find a Deep Structure of Estonian Legislation. Katie M. Atkinson (Toim.). Legal Knowledge and Information Systems - JURIX 2011: The Twenty-Fourth Annual Conference (93 - 102). IOS Press

Täks, E.; Lohk, A. (2010). An Alternative Method for Computerized Legal Text Restructuring. Radboud G. F. Winkels (Toim.). Legal Knowledge and Information Systems - JURIX 2010: The Twenty-Third Annual Conference (171 - 174). IOS Press

Lohk, A.; Võhandu, L. (2010). On the Estonian Emotion Vocabulary as It Is Presented in the Dictionary of Synonyms. In: Book of Abstracts: International Workshop "Emotions in and Around Language", Tallinn, 23-24. Sept. 2010. Tallinn, pp. 50.

7. Kaitstud lõputööd

„Laiendatud tekstiotsingu võimalused Piibli näitel: otsingufunktsionaalsuste määratlemine ja ettevalmistavad tööd“
Tehnikateaduste magister (Informaatika), Tallinna Tehnikaülikool, 2008

„Objektorienteeritud modelleerimise kasutamine inseneritarkvara loomisel“
Tehnikateaduste bakalaureus (Ehitustehnika), Tallinna Tehnikaülikool, 2003

8. Juhendatud lõputööd

Kaur Laanemäe, Meetod uudisvoogude tonaalsuse analüüsiks, Magistritöö, 2015

9. Muu tegevus

2012 – ... Eesti Rakenduslingvistika Aastaraamatu retsenseerimine

DISSERTATIONS DEFENDED AT TALLINN UNIVERSITY OF TECHNOLOGY ON *INFORMATICS AND SYSTEM ENGINEERING*

1. **Lea Elmik**. Informational Modelling of a Communication Office. 1992.

2. **Kalle Tammemäe**. Control Intensive Digital System Synthesis. 1997.

3. **Eerik Lossmann**. Complex Signal Classification Algorithms, Based on the Third-Order Statistical Models. 1999.

4. **Kaido Kikkas**. Using the Internet in Rehabilitation of People with Mobility Impairments – Case Studies and Views from Estonia. 1999.

5. **Nazmun Nahar**. Global Electronic Commerce Process: Business-to-Business. 1999.

6. **Jevgeni Riipulk**. Microwave Radiometry for Medical Applications. 2000.

7. **Alar Kuusik**. Compact Smart Home Systems: Design and Verification of Cost Effective Hardware Solutions. 2001.

8. **Jaan Raik**. Hierarchical Test Generation for Digital Circuits Represented by Decision Diagrams. 2001.

9. **Andri Riid**. Transparent Fuzzy Systems: Model and Control. 2002.

10. **Marina Brik**. Investigation and Development of Test Generation Methods for Control Part of Digital Systems. 2002.

11. **Raul Land**. Synchronous Approximation and Processing of Sampled Data Signals. 2002.

12. **Ants Ronk**. An Extended Block-Adaptive Fourier Analyser for Analysis and Reproduction of Periodic Components of Band-Limited Discrete-Time Signals. 2002.

13. **Toivo Paavle**. System Level Modeling of the Phase Locked Loops: Behavioral Analysis and Parameterization. 2003.

14. **Irina Astrova**. On Integration of Object-Oriented Applications with Relational Databases. 2003.

15. **Kuldar Taveter**. A Multi-Perspective Methodology for Agent-Oriented Business Modelling and Simulation. 2004.

16. **Taivo Kangilaski**. Eesti Energia käiduhaldussüsteem. 2004.

17. **Artur Jutman**. Selected Issues of Modeling, Verification and Testing of Digital Systems. 2004.

18. **Ander Tenno**. Simulation and Estimation of Electro-Chemical Processes in Maintenance-Free Batteries with Fixed Electrolyte. 2004.

19. **Oleg Korolkov**. Formation of Diffusion Welded Al Contacts to Semiconductor Silicon. 2004.

20. **Risto Vaarandi**. Tools and Techniques for Event Log Analysis. 2005.

21. **Marko Koort**. Transmitter Power Control in Wireless Communication Systems. 2005.

22. **Raul Savimaa**. Modelling Emergent Behaviour of Organizations. Time-Aware, UML and Agent Based Approach. 2005.

23. **Raido Kurel**. Investigation of Electrical Characteristics of SiC Based Complementary JBS Structures. 2005.

24. **Rainer Taniloo**. Ökonoomsete negatiivse diferentsiaaltakistusega astmete ja elementide disainimine ja optimeerimine. 2005.

25. **Pauli Lallo**. Adaptive Secure Data Transmission Method for OSI Level I. 2005.

26. **Deniss Kumlander**. Some Practical Algorithms to Solve the Maximum Clique Problem. 2005.

27. **Tarmo Veskioja**. Stable Marriage Problem and College Admission. 2005.

28. **Elena Fomina**. Low Power Finite State Machine Synthesis. 2005.

29. **Eero Ivask**. Digital Test in WEB-Based Environment 2006.

30. **Виктор Войтович**. Разработка технологий выращивания из жидкой фазы эпитаксиальных структур арсенида галлия с высоковольтным p-n переходом и изготовления диодов на их основе. 2006.

31. **Tanel Alumäe**. Methods for Estonian Large Vocabulary Speech Recognition. 2006.

32. **Erki Eessaar**. Relational and Object-Relational Database Management Systems as Platforms for Managing Softwareengineering Artefacts. 2006.

33. **Rauno Gordon**. Modelling of Cardiac Dynamics and Intracardiac Bioimpedance. 2007.

34. **Madis Listak**. A Task-Oriented Design of a Biologically Inspired Underwater Robot. 2007.

35. **Elmet Orasson**. Hybrid Built-in Self-Test. Methods and Tools for Analysis and Optimization of BIST. 2007.

36. **Eduard Petlenkov**. Neural Networks Based Identification and Control of Nonlinear Systems: ANARX Model Based Approach. 2007.

37. **Toomas Kirt**. Concept Formation in Exploratory Data Analysis: Case Studies of Linguistic and Banking Data. 2007.

38. **Juhan-Peep Ernits**. Two State Space Reduction Techniques for Explicit State Model Checking. 2007.

39. **Innar Liiv**. Pattern Discovery Using Seriation and Matrix Reordering: A Unified View, Extensions and an Application to Inventory Management. 2008.

40. **Andrei Pokatilov**. Development of National Standard for Voltage Unit Based on Solid-State References. 2008.

41. **Karin Lindroos**. Mapping Social Structures by Formal Non-Linear Information Processing Methods: Case Studies of Estonian Islands Environments. 2008.

42. **Maksim Jenihhin**. Simulation-Based Hardware Verification with High-Level Decision Diagrams. 2008.

43. **Ando Saabas**. Logics for Low-Level Code and Proof-Preserving Program Transformations. 2008.

44. **Ilja Tšahhirov**. Security Protocols Analysis in the Computational Model – Dependency Flow Graphs-Based Approach. 2008.

45. **Toomas Ruuben**. Wideband Digital Beamforming in Sonar Systems. 2009.

46. **Sergei Devadze**. Fault Simulation of Digital Systems. 2009.

47. **Andrei Krivošei**. Model Based Method for Adaptive Decomposition of the Thoracic Bio-Impedance Variations into Cardiac and Respiratory Components. 2009.

48. **Vineeth Govind**. DfT-Based External Test and Diagnosis of Mesh-like Networks on Chips. 2009.

49. **Andres Kull**. Model-Based Testing of Reactive Systems. 2009.

50. **Ants Torim**. Formal Concepts in the Theory of Monotone Systems. 2009.

51. **Erika Matsak**. Discovering Logical Constructs from Estonian Children Language. 2009.

52. **Paul Annus**. Multichannel Bioimpedance Spectroscopy: Instrumentation Methods and Design Principles. 2009.

53. **Maris Tõnso**. Computer Algebra Tools for Modelling, Analysis and Synthesis for Nonlinear Control Systems. 2010.

54. **Aivo Jürgenson**. Efficient Semantics of Parallel and Serial Models of Attack Trees. 2010.

55. **Erkki Joasoon**. The Tactile Feedback Device for Multi-Touch User Interfaces. 2010.

56. **Jürgo-Sören Preden**. Enhancing Situation – Awareness Cognition and Reasoning of Ad-Hoc Network Agents. 2010.

57. **Pavel Grigorenko**. Higher-Order Attribute Semantics of Flat Languages. 2010.

58. **Anna Rannaste**. Hierarcical Test Pattern Generation and Untestability Identification Techniques for Synchronous Sequential Circuits. 2010.

59. **Sergei Strik**. Battery Charging and Full-Featured Battery Charger Integrated Circuit for Portable Applications. 2011.

60. **Rain Ottis**. A Systematic Approach to Offensive Volunteer Cyber Militia. 2011.

61. **Natalja Sleptšuk**. Investigation of the Intermediate Layer in the Metal-Silicon Carbide Contact Obtained by Diffusion Welding. 2011.

62. **Martin Jaanus**. The Interactive Learning Environment for Mobile Laboratories. 2011.

63. **Argo Kasemaa**. Analog Front End Components for Bio-Impedance Measurement: Current Source Design and Implementation. 2011.

64. **Kenneth Geers**. Strategic Cyber Security: Evaluating Nation-State Cyber Attack Mitigation Strategies. 2011.

65. **Riina Maigre**. Composition of Web Services on Large Service Models. 2011.

66. **Helena Kruus**. Optimization of Built-in Self-Test in Digital Systems. 2011.

67. **Gunnar Piho**. Archetypes Based Techniques for Development of Domains, Requirements and Sofware. 2011.

68. **Juri Gavšin**. Intrinsic Robot Safety Through Reversibility of Actions. 2011.

69. **Dmitri Mihhailov**. Hardware Implementation of Recursive Sorting Algorithms Using Tree-like Structures and HFSM Models. 2012.

70. **Anton Tšertov**. System Modeling for Processor-Centric Test Automation. 2012.

71. **Sergei Kostin**. Self-Diagnosis in Digital Systems. 2012.

72. **Mihkel Tagel**. System-Level Design of Timing-Sensitive Network-on-Chip Based Dependable Systems. 2012.

73. **Juri Belikov**. Polynomial Methods for Nonlinear Control Systems. 2012.

74. **Kristina Vassiljeva**. Restricted Connectivity Neural Networks based Identification for Control. 2012.

75. **Tarmo Robal**. Towards Adaptive Web – Analysing and Recommending Web Users` Behaviour. 2012.

76. **Anton Karputkin**. Formal Verification and Error Correction on High-Level Decision Diagrams. 2012.

77. **Vadim Kimlaychuk**. Simulations in Multi-Agent Communication System. 2012.

78. **Taavi Viilukas**. Constraints Solving Based Hierarchical Test Generation for Synchronous Sequential Circuits. 2012.

79. **Marko Kääramees**. A Symbolic Approach to Model-based Online Testing. 2012.

80. **Enar Reilent**. Whiteboard Architecture for the Multi-agent Sensor Systems. 2012.

81. **Jaan Ojarand**. Wideband Excitation Signals for Fast Impedance Spectroscopy of Biological Objects. 2012.

82. **Igor Aleksejev**. FPGA-based Embedded Virtual Instrumentation. 2013.

83. **Juri Mihhailov**. Accurate Flexible Current Measurement Method and its Realization in Power and Battery Management Integrated Circuits for Portable Applications. 2013.

84. **Tõnis Saar**. The Piezo-Electric Impedance Spectroscopy: Solutions and Applications. 2013.

85. **Ermo Täks**. An Automated Legal Content Capture and Visualisation Method. 2013.

86. **Uljana Reinsalu**. Fault Simulation and Code Coverage Analysis of RTL Designs Using High-Level Decision Diagrams. 2013.

87. **Anton Tšepurov**. Hardware Modeling for Design Verification and Debug. 2013.

88. **Ivo Müürsepp**. Robust Detectors for Cognitive Radio. 2013.

89. **Jaas Je□ov**. Pressure sensitive lateral line for underwater robot. 2013.

90. **Vadim Kaparin**. Transformation of Nonlinear State Equations into Observer Form. 2013.

92. **Reeno Reeder**. Development and Optimisation of Modelling Methods and Algorithms for Terahertz Range Radiation Sources Based on Quantum Well Heterostructures. 2014.

93. **Ants Koel**. GaAs and SiC Semiconductor Materials Based Power Structures: Static and Dynamic Behavior Analysis. 2014.

94. **Jaan Übi**. Methods for Coopetition and Retention Analysis: An Application to University Management. 2014.

95. **Innokenti Sobolev**. Hyperspectral Data Processing and Interpretation in Remote Sensing Based on Laser-Induced Fluorescence Method. 2014.

96. **Jana Toompuu**. Investigation of the Specific Deep Levels in *p-*, *i-* and *n-* Regions of GaAs p^+-pin-n^+ Structures. 2014.

97. **Taavi Salumäe**. Flow-Sensitive Robotic Fish: From Concept to Experiments. 2015.

98. **Yar Muhammad**. A Parametric Framework for Modelling of Bioelectrical Signals. 2015.

99. **Ago Mõlder**. Image Processing Solutions for Precise Road Profile Measurement Systems. 2015.

100. **Kairit Sirts**. Non-Parametric Bayesian Models for Computational Morphology. 2015.

101. **Alina Gavrijaševa**. Coin Validation by Electromagnetic, Acoustic and Visual Features. 2015.

102. **Emiliano Pastorelli**. Analysis and 3D Visualisation of Microstructured Materials on Custom-Built Virtual Reality Environment. 2015.

103. **Asko Ristolainen**. Phantom Organs and their Applications in Robotic Surgery and Radiology Training. 2015.

104. **Aleksei Tepljakov**. Fractional-order Modeling and Control of Dynamic Systems. 2015.