

Concept Neighbourhoods in Lexical Databases

Uta Priss, L. John Old

Edinburgh Napier University, School of Computing,
www.upriss.org.uk, j.old@napier.ac.uk

Abstract. This paper discusses results from an experimental study of concept neighbourhoods in WordNet and Roget's Thesaurus. The general aim of this research is to determine ways in which neighbourhood lattices can be derived in real time from a lexical database and displayed on the web. In order to be readable the lattices must not be too large, not contain overlapping concepts or labels and must be calculated within seconds. Lattices should, furthermore, not be too small and they should contain sufficient complexity to be interesting for the viewer. For these purposes the sizes of the lattices of different types of concept neighbourhoods have been calculated. Using the size information should help with the task of on-line generation of the lattices.

1 Introduction

Concept neighbourhoods are a means of extracting smaller-sized formal contexts from a larger formal context whose concept lattice is too large to be viewed as a whole. The corresponding *neighbourhood lattices* consist of a concept and its neighbours. Roget's Thesaurus (RT) is an example for which the extraction of concept neighbourhoods has been studied in some detail (Priss & Old, 2004 and 2006). An on-line interface at www.roget.org lets users explore concept neighbourhoods of Roget's Thesaurus in real-time. The algorithm for constructing the neighbourhoods uses a number of heuristics which ensure that the lattices are neither oversized, nor trivial.

The goal of our current research is to implement a similar interface for WordNet (Fellbaum, 1998) and potentially for other lexical databases in the future. Previous research has shown that the formation of concept neighbourhoods and the establishment of heuristics for generating reasonably-sized lattices depend on the structure of the underlying resources. For example, Dyvik's (2004) method for constructing a thesaurus from a bilingual corpus (which is very similar to our method of building concept neighbourhoods) does not work so well if a bilingual dictionary is used instead of a corpus (Priss & Old, 2005). The reason for this is that the translational relations between words in a corpus show more diversity than in a dictionary. Thus, even if a lexical database has a similar hierarchical structure to Roget's Thesaurus, the algorithms for forming concept neighbourhoods may require some adjustment. Furthermore, if a database such as WordNet contains a variety of semantic and lexical relations, it seems reasonable to attempt to incorporate these existing relations into the formation of concept neighbourhoods.

WordNet (Fellbaum, 1998) is a lexical database which groups words into *synsets* of synonyms (or near synonyms). Each synset belongs to a part of speech (noun, verb,

adjective, adverb) and can participate in several part-of-speech-dependent semantic and lexical relations. For example, the semantic relations for nouns are IS-A relations (hyponymy/hyponymy) and several types of part-whole relations (meronymy). In contrast to semantic relations which are defined between synsets, lexical relations (such as antonymy) are defined between words. This is because the designers of WordNet took a psychological perspective and decided that antonymy is dependent on word associations. For example, in the synset “large, big”, “large” is an antonym of “small” and “big” is an antonym of “little” because people tend to associate these. From a logical perspective it can be argued that antonymy simply expresses a form of contrast or opposition that can be applied to the whole synset. Therefore in our applications we sometimes generalise a lexical relation (between two words from different synsets) into a semantic relation (between all words of the two synsets). Alternatively it is also possible to treat a semantic relation between two synsets as a lexical relation between all words of the two synsets.

WordNet has been used in many research projects and has been visualised in a number of formats. Probably the most well-known visualisation is the one at visualthesaurus.com, which employs Java-applets to draw networks around words using WordNet’s semantic relations and the spring-embedder algorithm. This visualisation differs from our research because it does not result in lattices, but in networks, which completely ignore the hierarchical structure of WordNet’s relations. WordNet has been used in several Formal Concept Analysis projects for example by Hotho et al. (2003) as a means for improving text clustering by exploiting WordNet’s semantic relations and by Martin & Eklund (2005) for adding hypernymic concepts in a lattice of a semantic file system. But in these projects both WordNet and FCA are just tools used for other purposes and, again, this kind of research differs from what we are intending to do.

Also of interest are studies that compare WordNet and Roget’s Thesaurus (for example, Old (2003)). Kennedy & Szpakowicz (2008) discover that different editions of Roget’s Thesaurus are quite similar to each other and to WordNet with respect to the calculation of semantic similarity and their usage in language-based information retrieval enhancement methods. Therefore it should be expected that concept neighbourhoods extracted from Roget’s Thesaurus and WordNet are similar in structure and size.

Section 2 discusses requirements of an on-line interface for concept neighbourhoods. Section 3 introduces the notions of concept neighbourhoods and neighbourhood lattices in more detail. Section 4 presents examples of concept neighbourhoods from WordNet. Section 5 discusses experimental results that were conducted on the WordNet database and in comparison with Roget’s Thesaurus.

2 An on-line interface for concept neighbourhoods

Creating an on-line interface that generates concept neighbourhoods on the fly poses a number of challenges. Both WordNet and Roget’s Thesaurus (RT) contain more than 100,000 words. It would therefore be inefficient to generate all neighbourhood lattices in advance and store them as image files. Furthermore, an interface should allow for a certain amount of flexibility. It should offer choices of semantic relations to be included, degrees of size restrictions, and so on, because there are many possibilities for creating

concept neighbourhoods, and users may have different interests and preferences. The idea is to create an interface similar to Dyvik's (2004) semantic mirrors interface, which allows users to choose different data sources, thresholds and limits.

The technology we are using at the moment consists of MySQL databases of WordNet and RT and the FcaStone¹ software, which uses Graphviz² for computing the graph layouts. The database for RT was created by the second author (based on work described by Sedelow & Sedelow (1993)). The WordNet database is built using WordNet SQL Builder³.

Requirements for generating on-line lattices are that the lattices should:

- be easy to read, not too large, and not too complex so that they are viewable without major zooming or scrolling;
- contain no overlapping nodes or labels;
- be generated within seconds.

Addressing the last requirement first: FcaStone's performance declines sharply when one tries to compute lattices with more than 100 concepts, but smaller lattices can be computed within seconds. Because lattices with more than 50 concepts are not very readable anyway, the main limit for the size of the lattices is readability, not software performance.

Nodes and labels should not overlap. An automatically generated layout of a lattice needs to determine the placement of the concepts, and also the placement of the labels. Since Graphviz does not provide a separate option for placing labels, we are representing each concept as a box which contains the attributes in the top half and the objects in the bottom half (as in Figure 1). Both attributes and objects are limited to 30 characters. If there are more objects or attributes, the string is truncated at the 30th character and dots (...) are inserted at the end. In that way, it can be guaranteed that neither concepts, nor labels overlap anywhere.

To some degree the readability depends on the structure of the lattice. Because of the placement of the labels inside the concept boxes, only up to 10 concepts (or less in smaller browser windows) can be displayed side by side. Graphviz sometimes draws the edges slightly curved, not straight, and in longer anti-chains and crowns, the edges may touch each other or overlap and become difficult to trace. Thus, lattices with the same number of concepts can have very different readability: a lattice with 20 concepts which are arranged in 4-5 levels may be easier to read than a lattice with 20 concepts which contains an anti-chain with more than 10 concepts.

3 Concept Neighbourhoods and Neighbourhood Lattices

A concept neighbourhood is extracted by starting with one item (word, term or concept) and then retrieving other items the first item is related to and so on. This is called the *plus operator* (Priss & Old, 2004) and is usually applied to a formal context of objects

¹ <http://fcastone.sourceforge.net>

² <http://www.graphviz.org>

³ <http://wnsqlbuilder.sourceforge.net>

and attributes, such as words and their translations into another language, words and their sense numbers in Roget’s Thesaurus, or documents and their classification codes. The complete formal context of a lexical database might contain more than 100,000 rows and columns. This is too large to build a readable lattice diagram. Starting with one object or attribute the plus operator is usually applied for a fixed number of times, because if the plus operator is applied an unlimited number of times the neighbourhood might grow until it encompasses the complete or nearly complete lexical database. Apart from stopping the plus operator after a fixed number of steps, it is also possible to apply several restriction methods (Priss & Old, 2004) in order to prevent the concept neighbourhoods from becoming too large.

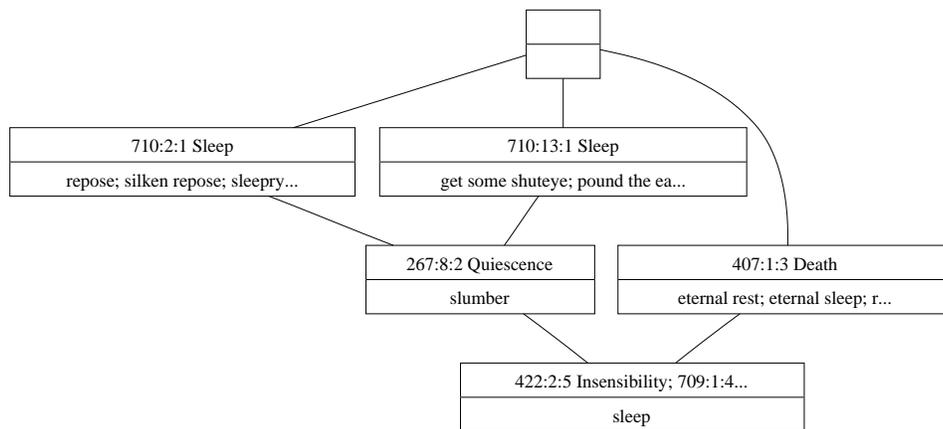


Fig. 1. A neighbourhood lattice in RT for the word “sleep”

A plain n - m -neighbourhood starts with an object and has the plus operator applied $(2n - 2)$ -times to obtain the set of objects and $(2m - 1)$ -times to obtain the set of attributes. Thus, a 2-1-neighbourhood of WordNet or RT consists of all the words in the synsets of a word and the senses of the original word. A 2-2-neighbourhood consists of all the words in the synsets of a word and all of their senses. Figure 1 shows a 2-1-neighbourhood lattice for the word “sleep” in RT. The plus operator was applied twice: first, to find all the senses of “sleep” (as formal attributes) and then one more time to find other words (as formal objects) which have the same senses. The senses are described numerically as “Category number:Paragraph number:Synset number” followed by the head word of the category. For example, “710:2:1” and “710:13:1” are two senses, both belonging to the category “Sleep”. The layout and design of the lattice was generated automatically as described in the previous section.

4 Concept Neighbourhoods in WordNet

WordNet contains many different types of relations. Therefore there are many possibilities for creating different types of neighbourhood lattices. One possibility is to ignore the semantic relations completely and simply generate plain neighbourhoods exactly as in RT by taking a word, its synsets, all other words in these synsets, their other synsets and so on.

Figure 2 shows the 2-1 neighbourhood for “sleep” in WordNet. The senses are represented by their index numbers. The fact that both RT (in Figure 1) and WordNet contain the words “eternal rest, eternal sleep” in the same sequence in the neighbourhood of “sleep” is not a coincidence because when WordNet was started, the WordNet lexicographers used RT as one of their sources. Thus, WordNet was influenced by RT. Otherwise, the two neighbourhoods only share the words “rest” and “slumber”. In other examples we looked at, it also seemed to be the case that only very few words are shared between the WordNet and RT neighbourhoods. We have not yet explored this more systematically, but it might be of interest to calculate the intersection between the WordNet and RT neighbourhoods because they might highlight core synonyms of each word. But that is left for future research.

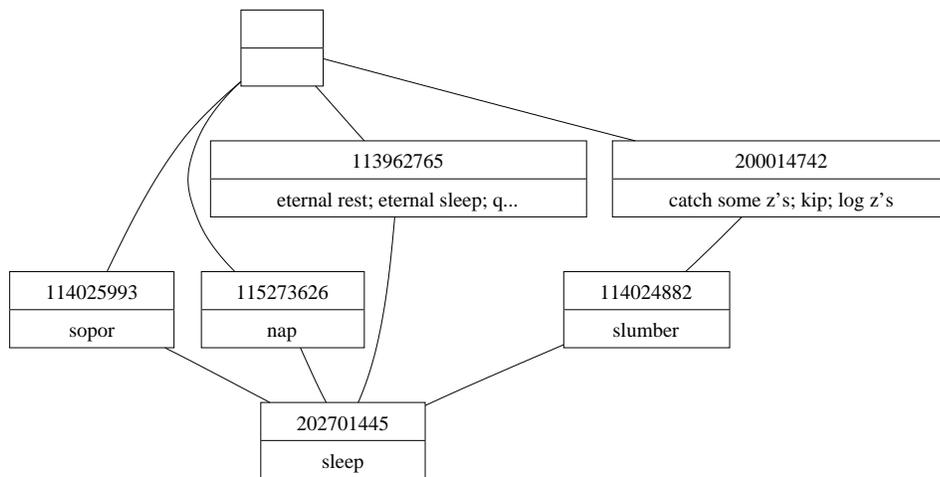


Fig. 2. A lattice of a plain 2-1 neighbourhood in WordNet for the word “sleep”

A 2-1 neighbourhood will only have some interesting structure if the synsets (representing the different senses of a word) have intersections of more than one word. In RT, synsets seem to be fairly large and the synsets corresponding to different senses tend to have larger intersections. In WordNet, however, the synsets tend to be smaller and tend to intersect only in one, sometimes two words. In Figure 2, the only intersection other than “sleep” is the word “slumber” which occurs in two synsets. The reason for this difference between WordNet and RT may be that the lexicographers who create WordNet

can view all synsets (senses) of a word when they edit the data. Synsets can be created in a similar manner as traditional dictionaries where the senses of a word are carefully distinguished and balanced. It may be that too much overlap between synsets is deliberately avoided by WordNet's lexicographers. In Roget's Thesaurus, the different senses of a word are only visible in the index, which was created (for the original edition) after the construction of the thesaurus was finished. Roget would not have been able to deliberately control the overlap between synsets. Therefore 2-1-neighbourhoods of polysemous words (especially nouns) in WordNet are often anti-chains whereas in RT their 2-1-neighbourhoods tend to have more interesting structures. For WordNet other means of creating neighbourhood lattices need to be investigated.

Another possibility is to incorporate WordNet's semantic relations into the building of neighbourhood lattices. One difficulty with this approach is that all parts of speech in WordNet have different types of semantic relations and require different approaches. For nouns and verbs the hypernymy relation can be used as follows: the words of the synsets belonging to all senses of a word are taken as formal objects and the words of the hypernymic synsets are the formal attributes. The relation between objects and attributes is the semantic relation between synsets and their hypernymic synsets but treated as a lexical relation between the words in the synsets. We call this the *hypernymy neighbourhood* because it is based on the hypernymy relation. A *hyponymy neighbourhood* is formed by using hyponymy, and so on. Adjectives and adverbs do not have a hypernymy relation in WordNet, but their "similarity" relation can be used in the same manner.

Figure 3 shows an example of a hypernymy neighbourhood of "sleep" in WordNet. Figure 4 displays the corresponding synsets and their hypernymy relation in WordNet. The example shows that most synsets are maintained as extensions or intensions in a hypernymy neighbourhood. The extensions/intensions are only different from WordNet synsets if the synsets share words (as discussed above for plain neighbourhoods) or share hypernyms. For example, in Figure 3, the left-most concept has the extension "sopor, slumber, sleep" which is not a synset in WordNet. This extension emerges because the two left-most synsets in Figure 4 have the same hypernymic synset. In our opinion, WordNet synsets that share words or hypernyms exhibit implicit structures (in the sense of Sedelow's (1998) "inner structure analysis"), which are detected and visualised when forming hypernymy neighbourhoods. The lattice in Figure 3 is more similar to the RT lattice than the one in Figure 2. But according to our experimental results (see next section), hypernymy neighbourhoods are not in general more similar to RT lattices than plain 2-1 neighbourhood lattices.

Unfortunately, the hypernymy neighbourhoods of most nouns tend to be uninteresting because they tend to be fairly small and predominantly form anti-chains. Hypernymy neighbourhoods of verbs reveal more structures. The difference between nouns and verbs might be caused either by a structural difference or by the fact that two different lexicographers are responsible for nouns and verbs in WordNet who might use different strategies for implementing hypernymy relations.

Apart from synsets sharing words or hypernyms, it might also be interesting to identify words which occur both as objects and attributes in the same neighbourhood. In Figure 3, the word "rest" occurs both as an attribute and an object (under the attribute

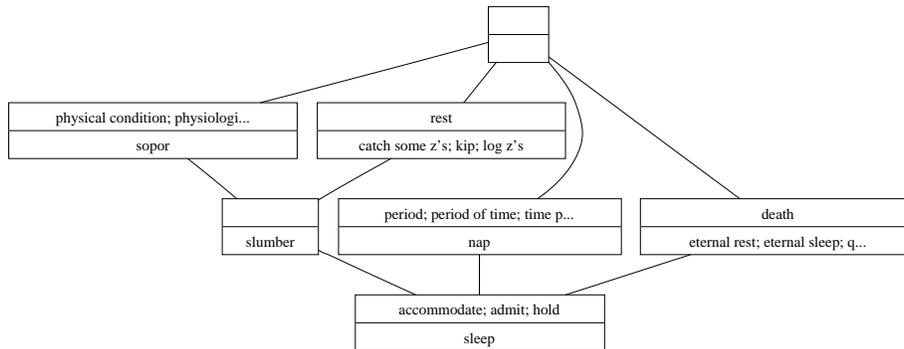


Fig. 3. A lattice of a hypernymy neighbourhood in WordNet

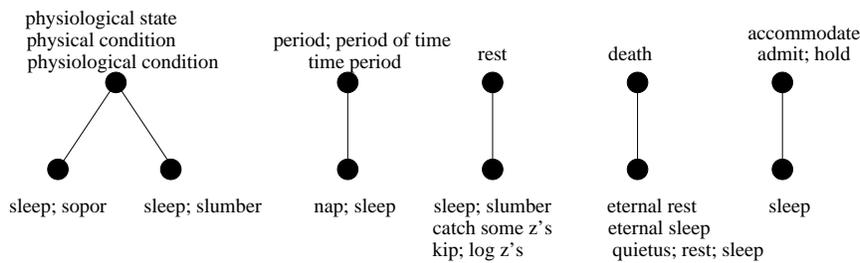


Fig. 4. WordNet's Hypernymy Relation

“death”, but not visible in the Figure because the objects are truncated to 30 characters). An identity relation can be added to the formal context which inserts a cross wherever an object equals an attribute. The result is shown in Figure 5. Another possibility would be to insert crosses whenever there is a substring match between the words in a neighbourhood (matching “rest” and “eternal rest”). These options need to be explored in more detail, but some preliminary analysis indicates that unfortunately, for nouns in WordNet, there is not a significant overlap between objects and attributes.

5 Experimental results

In order to obtain a better idea as to what kinds of neighbourhood lattices might be most promising for WordNet, we calculated the number of concepts for plain 2-1, plain 2-2 and hypernymy neighbourhoods in WordNet. For comparison we also calculated plain 2-1 and plain 2-2 neighbourhoods in RT. For our test data we chose a list of 45,000 frequently used words. From this list 26,000 words occur in WordNet and 21,000 in RT. The list includes names, placenames and so on, not all of which occur in WordNet and RT. Because the list does not include phrases or compound words, it contains only about 1/4 of the words in WordNet and RT. But phrases and compound words tend to

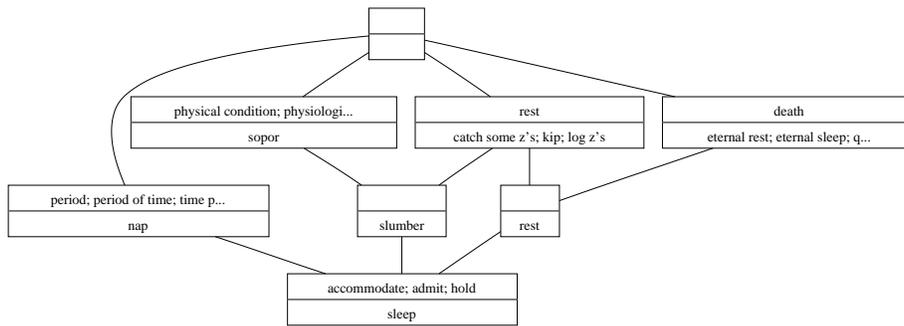


Fig. 5. A Neighbourhood Lattice with Identity in WordNet

be less polysemous and can be expected to generate smaller concept neighbourhoods. Thus, our test data contains all the interesting words. Because not all nouns and verbs have hypernyms in WordNet and adjectives and adverbs do not have hypernyms at all, only about 17,000 words are used for the hypernymy neighbourhoods.

Table 1 shows the words with the largest neighbourhood lattices in WordNet and RT. The words are sorted by their average ranking in WordNet hypernymy, plain 2-1 and plain 2-2 neighbourhoods. The largest hypernymy and plain 2-1 neighbourhood lattices contain about 40 concepts in WordNet and can be graphically displayed. The largest plain 2-2 neighbourhood lattice in WordNet (for the word “take”) contains 657 concepts. In the three types of concept neighbourhoods in WordNet that we looked at, the words tend to be ranked in similar positions.

Table 1 also shows the sizes of the RT plain 2-1 neighbourhood lattices of these words. There is slightly less agreement between the WordNet and RT rankings. The last column “top in both” indicates which words have the largest neighbourhood lattices both in WordNet and RT. The word “pass” is among the 5 largest lattices in all types of neighbourhoods. On the other hand, “turn” which has the second largest neighbourhood lattice in RT has much smaller neighbourhood lattices in WordNet.

A comparison with Old (2003, p. 183) shows that the words with the largest neighbourhood lattices also tend to be the most polysemous words, i.e. the ones with the most senses. This is of course not surprising because the senses are used for the construction of neighbourhoods. Although individual words can slightly differ with respect to the sizes of their neighbourhood lattices in WordNet and Roget, all of the words with large neighbourhood lattices have some verb senses and tend to be of Anglo-Saxon origin. This is true not just for the words in Table 1, but in general. Words which have only noun or adjective senses have much smaller neighbourhood lattices both in RT and WordNet.

Tables 2 and 3 show a listing just for adjectives. The rank in the tables is calculated as the average of (size of lattice) ÷ (max size of lattice in this neighbourhood type). It is interesting to observe that many negative adjectives have large neighbourhood lattices. The largest one in RT is “vile”; in WordNet “hard” and “grim”. Among the top ten largest 2-2 neighbourhood lattices for adjectives in RT are “abominable”, “obnoxious”,

Table 1. The words with the largest neighbourhood lattices in WordNet and RT and their number of concepts

	WN: hypernymy	WN: plain 2-1	WN: plain 2-2	RT: 2-1	top in both
pass	36	44	582	65	yes
break	39	46	543	44	
take	31	37	657	34	
run	34	39	497	79	yes
hold	31	39	490	47	
set	29	37	336	86	yes
draw	28	30	464	29	
check	27	31	469	52	
make	23	32	549	34	
get	26	28	530	46	
cut	28	36	296	118	yes
go	22	30	561	55	yes
give	26	28	469	17	
place	28	28	321	42	
point	28	29	302	51	
rise	40	42	236	40	
return	28	26	280	40	
turn	23	24	293	87	yes

“odious”, “contemptible” and “despicable”. For some reason their lattices are slightly smaller in the 2-1 neighbourhoods and therefore not in Table 2. The adjectives that have large lattices across all neighbourhood types tend to be short words of Anglo-Saxon origin.

Table 2. The adjectives with the largest neighbourhood lattices in RT

	RT: plain 2-1	RT: plain 2-2	rank
vile	48	227	1
fixed	43	201	0.89
soft	43	147	0.77
hard	32	179	0.73
easy	40	126	0.69
proper	32	152	0.67
sad	23	172	0.62

In addition to looking at the words with the largest neighbourhood lattices, we also looked at the size distributions. Figure 6 shows the number of lattices with up to 25 concepts for the three types of neighbourhoods in WordNet and the plain 2-1 neighbourhoods in RT. The number of lattices of size 1 is not a good indication of anything because as mentioned before the data sets used for the different tests have different

Table 3. The adjectives with the largest neighbourhood lattices in WordNet

	WN: plain 2-1	WN: plain 2-2	rank
hard	14	81	0.83
grim	9	122	0.82
easy	10	106	0.79
fresh	12	86	0.78
big	13	74	0.77
tight	11	86	0.75
soft	14	55	0.73
strong	9	78	0.64
awful	11	56	0.62
just	10	61	0.61
substantial	8	55	0.51

sizes. The set used for RT is smaller than the one used for WordNet. The hypernymy set is smallest because not all words in WordNet have hypernyms. Many of the words with lattice size 1 are proper nouns, abbreviations and other specialised terms. It is to be expected that names and proper nouns have small lattices. There are however some surprises. For example the plain 2-2 neighbourhood lattice of “Adam” in WordNet contains 40 concepts. The reason for this is that “Adam” is a synonym of “ecstasy” and “go” as a hyponym of the drug “mdma”. Because “go” is very polysemous, the 2-2 neighbourhood for “Adam” is large too. But such kind of effects are anomalies and indicate homographic or metaphoric word use.

Figure 6 shows that lattices are largest using the plain 2-2 neighbourhoods (which is not surprising because the plus operator is used one more time). The plain 2-1 lattices in RT are larger than in WordNet. The hypernymy neighbourhood lattices are smallest. Figure 6 indicates that the size of the lattices has an impact on the distribution. Normally one might expect to see some kind of power law distribution, which is common for linguistic phenomena (and has been shown to apply to neighbourhood closure lattices in RT by Priss & Old (2006)). But lattices of size 3 are much rarer than lattices of size 2 or 4. This is because as mentioned above many of the lattices have the shape of an anti-chain or crown. But the only possibility to form a lattice with 3 concepts is as a chain, which corresponds to a subset relation among the synsets and is very rare.

Table 4 shows the percentages of lattices with fewer than 6 concepts, between 6 and 45, and more than 45 concepts. Having fewer than 6 concepts is undesirable because such lattices may not be very interesting. On the other hand, for specialised terms, proper nouns, and so on, it may be unavoidable to have a small lattice. Lattices with 6 to 45 concepts should be viewable. If 45 concepts is too large, restriction can be applied (Priss & Old, 2004), which removes most concepts that are meet- and join-irreducible and reduces the number of concepts without changing the core structure of the lattice. Lattices with more than 45 concepts are most likely not readable and require restriction.

Table 4 indicates that in the case of WordNet a good strategy might be to create plain 2-2 neighbourhoods for most words except for the 6% for which the lattices have more than 45 concepts. For such words, hypernymy or plain 2-1 neighbourhoods should be

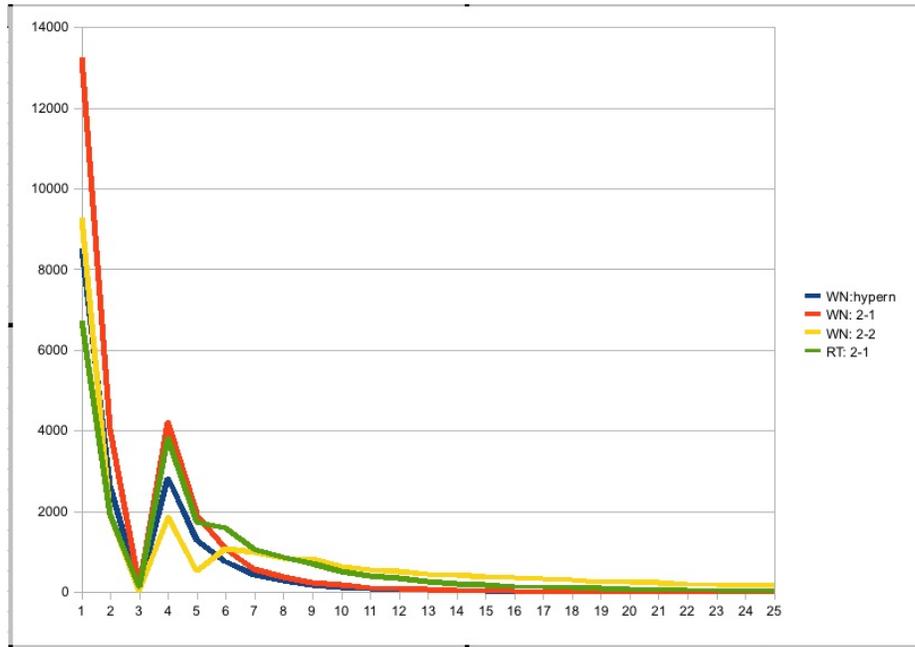


Fig. 6. Number of lattices with up to 25 concepts

chosen, which are guaranteed to have smaller lattices. It should be noted that this does not imply that the hypernymy relation should never be used. It is possible to combine the hypernymy and plain 2-2 neighbourhoods. Also different strategies might be necessary for the different parts of speech. For RT, plain 2-2 neighbourhoods are suitable for 68% of the words, whereas plain 2-1 neighbourhoods should be used for 31% of the words. A restricted plain 2-1 neighbourhood should be used for the remaining 1%.

From an implementation viewpoint, the sizes of the lattices should be stored in a look-up table. This can be calculated while the database is off-line, so it does not matter if this is slow. We have not yet calculated the sizes for all words in WordNet and RT, but as mentioned before, the entries which have not been included in our test data tend to be phrases and compound words which tend to have smaller lattices. Thus, most likely our list of the 6% of words with large plain 2-2 neighbourhoods is a complete or nearly complete listing for WordNet. Our list of 31% of the words for RT will also be nearly complete. Our previous approach to limiting the size of the lattices with respect to our on-line RT interface has been to use heuristics derived from the number of objects and attributes of the context. Using a look-up table instead of heuristics seems to be a better approach because it is more precise while still requiring about the same amount of computational resources.

Table 4. The percentages of lattices with different sizes

	WN:hypern	WN: 2-1	WN: 2-2	RT: 2-1
smaller than 6	90%	91%	52%	68%
between 6 and 45	10%	9%	42%	31%
larger than 45	0%	0%	6%	1%

6 Conclusion

The main result of this paper is that we have extended our concept neighbourhood modelling from Roget's Thesaurus to WordNet. We have conducted a number of experiments, both with respect to looking at individual examples of words, but also obtaining a distribution of the sizes of neighbourhood lattices of different types. The results show that there are differences between parts of speech and between WordNet and RT. A feasible approach to determining in advance which type of neighbourhood to use for which word appears to be to calculate the sizes of the neighbourhood lattices once, and store them as a look-up table in the database. We have not yet calculated these numbers for all words and all types of semantic relations but it appears that words with large lattices in one type also have large lattices in other types. Verbs of Anglo-Saxon origin tend to have the largest lattices. The data about the lattice sizes which we have calculated so far appears sufficient to prevent the construction of oversized neighbourhood lattices. Users of the on-line interface⁴ can still be given some flexibility to experiment with different types of neighbourhood lattices. Only the construction of oversized lattices needs to be avoided.

It might be of linguistic interest to conduct a more detailed analysis of the data that we have collected so far. It might indicate a classification of words according to their types of neighbourhoods or offer more insights into the structure of synonymy and other semantic relations. Also, if a word behaves differently across different neighbourhoods, such anomalies might highlight interesting facts or errors in the data. But such analyses are left for future research.

Other plans for future research include: extending this research to other lexical databases; investigating the use of faster algorithms; and improving and testing the user interface of our website. A further idea is to investigate whether there are other indicators than number of concepts that can be used to determine how readable the lattices are. For example, for the same number of concepts, lattices that have about the same width as height might be more readable than lattices that are "short" and "wide". Our experimental data suggests that "tall" and "narrow" lattices are unlikely with respect to WordNet and RT. One indicator for width might be the degree of sparseness of the formal context.

⁴ The interface for RT is currently at <http://www.ketlab.org.uk/roget.html>. The WordNet interface will also be added to that site in the near future.

References

1. Dyvik, H. (2004). Translations as semantic mirrors: from parallel corpus to wordnet. *Language and Computers*, Vol. 49, iss. 1, Rodopi, p. 311-326.
2. Fellbaum, Christiane (ed.) (1998). *WordNet - An Electronic Lexical Database*. MIT Press.
3. Hotho, A.; Staab, S.; Stumme, G. (2003). Explaining text clustering results using semantic structures. In: *Principles of Data Mining and Knowledge Discovery*, 7th European Conference, PKDD 2003, Croatia. LNCS. Springer, p. 22-26.
4. Kennedy, A.; Szpakowicz, S. (2008). Evaluating Roget's Thesauri. *Proc. of ACL-08: HLT*, Columbus Ohio, USA Association for Computational Linguistics 416-424.
5. Martin, Ben; Eklund, Peter (2005). Applying Formal Concept Analysis to Semantic File Systems Leveraging Wordnet. *Proceedings of the 10th Australasian Document Computing Symposium*.
6. Old, L. John (2003). *The Semantic Structure of Roget's, A Whole-Language Thesaurus*. PhD Dissertation. Indiana University.
7. Priss, Uta; Old, L. John (2004). Modelling Lexical Databases with Formal Concept Analysis. *Journal of Universal Computer Science*, Vol 10, 8, p. 967-984.
8. Priss, Uta; Old, L. John (2005). Conceptual Exploration of Semantic Mirrors. In: Ganter; Godin (eds.), *Formal Concept Analysis: Third International Conference, ICFCA 2005*, Springer Verlag, LNCS 3403, p. 21-32.
9. Priss, Uta; Old, L. John (2006). An application of relation algebra to lexical databases. In: Schaerfe, Hitzler, Ohrstrom (eds.), *Conceptual Structures: Inspiration and Application*, Proceedings of the 14th International Conference on Conceptual Structures, ICCS'06, Springer Verlag, LNAI 4068, p. 388-400.
10. Sedelow, S.; Sedelow, W. (1993). The Concept concept. *Proceedings of the Fifth International Conference on Computing and Information*, Sudbury, Ontario, Canada, p. 339-343.
11. Sedelow, Walter A., Jr. (1988). Computer-based planning technology: an overview of inner structure analysis. *Sixth Annual Conference on New Technology and Higher Education: Acquisition, Integration, and Utilization*.