Evaluation of automatic updates of *Roget's Thesaurus*

Alistair Kennedy[1] *and Stan Szpakowicz*[2,1]
[1] School of Electrical Engineering and Computer Science
University of Ottawa, Ottawa, Ontario, Canada
[2] Institute of Computer Science
Polish Academy of Sciences, Warsaw, Poland

ABSTRACT

Thesauri and similarly organised resources attract increasing interest of Natural Language Processing researchers. Thesauri age fast, so there is a constant need to update their vocabulary. Since a manual update cycle takes considerable time, automated methods are required. This work presents a tuneable method of measuring semantic relatedness, trained on *Roget's Thesaurus*, which generates lists of terms related to words not yet in the *Thesaurus*. Using these lists of terms, we experiment with three methods of adding words to the *Thesaurus*. We add, with high confidence, over 5500 and 9600 new word senses to versions of *Roget's Thesaurus* from 1911 and 1987 respectively.

We evaluate our work both manually and by applying the updated thesauri in three NLP tasks: selection of the best synonym from a set of candidates, pseudo-word-sense disambiguation and SAT-style analogy problems. We find that the newly added words are of high quality. The additions significantly improve the performance of *Roget's*-based methods in these NLP tasks. The performance of our system compares favourably with that of *WordNet*-based methods. Our methods are general enough to work with different versions of *Roget's Thesaurus*.

Keywords:
lexical resources, Roget's Thesaurus, WordNet, semantic relatedness, synonym selection, pseudo-word-sense disambiguation, analogy

1 INTRODUCTION

Thesauri and other similarly organised lexical knowledge bases play a major role in applications of Natural Language Processing (NLP). While *Roget's Thesaurus*, whose original form is 160 years old, has been applied successfully, the NLP community turns most often to *WordNet* (Fellbaum 1998). *WordNet*'s intrinsic advantages notwithstanding, one of the reasons is that no other similar resource, including *Roget's Thesaurus*, has been publicly available in a suitable software package. It is, however, important to note that *WordNet* represents *one* of the methods of organising the English lexicon, and need not be the superior resource for every task. *Roget's Thesaurus* updated with the most recent vocabulary can become a competitive resource whose quality measures up to *WordNet*'s on a variety of NLP applications. In this paper, we describe and evaluate a few variations on an innovative method of updating the lexicon of *Roget's Thesaurus*.

Work on learning to construct or enhance a thesaurus by clustering related words goes back over two decades (Tsurumaru *et al.* 1986; Crouch 1988; Crouch and Yang 1992). Few methods use an existing resource in the process of updating that same resource. We employ *Roget's Thesaurus* in two ways when creating its updated versions. First, we construct a measure of semantic relatedness between terms, and tune a system to place a word in the *Thesaurus*. Next, we use the resource to "learn" how to place new words in the correct locations. This paper focusses on finding how to place a new word appropriately.

We evaluate our lexicon-updating methods on two versions of *Roget's Thesaurus*, with the vocabulary from 1911 and from 1987. Printed versions are periodically updated, but new releases – neither easily available to NLP researchers nor NLP-friendly – have had little effect on the community. The 1911 version of *Roget's Thesaurus* is freely available through Project Gutenberg.[1] We also work with the 1987 edition of *Penguin's Roget's Thesaurus* (Kirkpatrick 1987). An open Java API for the 1911 *Roget's Thesaurus* and its updated versions – including every addition we discuss in this paper – are available on the Web as the *Open Roget's Project.*[2] The API has been built on the work of Jarmasz (2003).

[1] `http://www.gutenberg.org/ebooks/22`

[2] `http://rogets.eecs.uottawa.ca/`

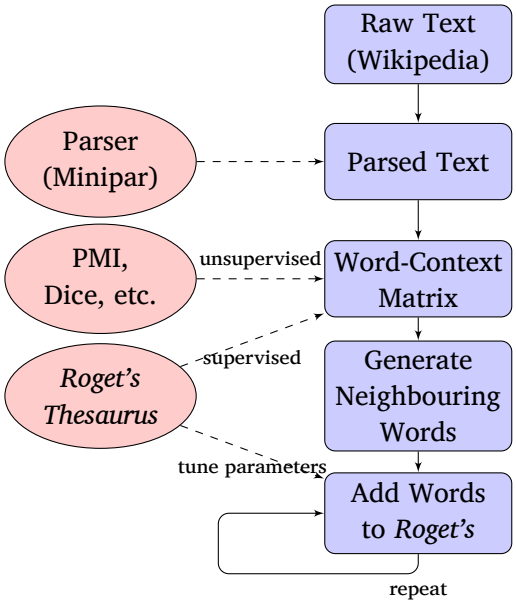


Figure 1:
The process of adding new words to *Roget's Thesaurus*

Figure 1 outlines the process of updating *Roget's Thesaurus*. We work with Wikipedia as a corpus and with the parser *MINIPAR* (Lin 1998a). Raw text is parsed, and a word–context matrix is constructed and re-weighted in both a supervised and an unsupervised manner. The nearest synonyms of each word in the matrix are generated and a location for them in *Roget's Thesaurus* is deduced using it as a source of tuning data. The last step can be applied iteratively to update the lexicon of *Roget's Thesaurus*.

This work makes six main contributions:

- apply the supervised measures of semantic relatedness from (Kennedy and Szpakowicz 2011) and (Kennedy and Szpakowicz 2012) to the updating of *Roget's Thesaurus*, and evaluate it carefully;
- propose and compare three methods of automatically adding words to *Roget's Thesaurus*;
- build the updated editions of the 1911 and 1987 versions of *Roget's Thesaurus*;
- create new datasets for pseudo-word-sense disambiguation and the selection of the best synonym;

- propose and evaluate a new method for solving SAT-style analogy problems;
- compare semantic similarity calculation with *Roget's Thesaurus* and *WordNet* on accuracy and on runtime.

1.1 *About Roget's Thesaurus*

In the early 1800s, Peter Mark Roget, a physician, began to categorise terms and phrases for his personal use in writing. The ensuing *Roget's Thesaurus*, first published in 1852, has gone through many revisions continuing to this day (Kendall 2008). A nine-level hierarchy makes up most of the structure of *Roget's Thesaurus*:

1	Class	6	Part of Speech
2	Section	7	Paragraph
3	Sub-Section	8	Semicolon Group
4	Head Group	9	Words and Phrases
5	Head		

Eight classes are subdivided into Sections and Sub-Sections. There are around 1000 Heads – the main category in *Roget's Thesaurus*, corresponding to major concepts. Heads with opposing or complementary concepts form a Head Group. A Part of Speech (POS) groups all noun/verb/adjective/adverb realisations of the Head's concept. The closest counterpart of *WordNet*'s synsets is a Semicolon Group (SG). An SG contains closely related words (usually near-synonyms); a Paragraph contains related SGs. Note the division by part-of-speech quite low in the hierarchy, not at the very top as in *WordNet*. We define a *Roget's grouping* to be the set of words contained within an instance of any of these levels. A Section or even a Class is also a *Roget's grouping*, but usually we talk about words in the same POS, Paragraph or SG.

Figure 2 shows an example of a Head. Head #586 in the 1911 *Roget's Thesaurus* contains terms pertaining to language. A number before a word refers to a Head in which that word-sense may also be found. Although a thorough update of *Roget's Thesaurus* should include such *cross-references*, they are beyond the scope of this work.[3]

[3] They do not figure in any of the applications we consider here to test the quality of the updated versions of the *Thesaurus*.

Class 5: Intellect: communication of ideas
Section 3: Means of communicating ideas
Sub-Section: Conventional means
Head Group: 586 Language
Head: 586 Language

N. *language*; *595* phraseology; *608* speech; tongue, lingo, vernacular; mother tongue, vulgar tongue, native tongue; household words; King's English, Queen's English; *589* dialect.

confusion of tongues, Babel, pasigraphie; *sign 576* pantomime; onomatopoeia; betacism, mimmation, myatism, nunnation; pasigraphy.

lexicology, philology, glossology, glottology; linguistics, chrestomathy; paleology, paleography; comparative grammar.

literature, letters, polite literature, belles lettres, muses, humanities, literae humaniores, republic of letters, dead languages, classics; genius of language; *scholar 516* scholarship.

VB. *592 express by words*.

ADJ. *lingual*, linguistic; dialectic; vernacular, current; bilingual; diglot, hexaglot, polyglot; literary.

Figure 2: Head 586: Language from the 1911 *Roget's Thesaurus*

1.2 *Where to add new words to Roget's Thesaurus*

The number of Heads and POSs per Head have changed little between the 1911 and 1987 versions of *Roget's Thesaurus*. We can aim to add new words in three different ways:

- in an existing SG,
- in a new SG in an existing Paragraph,
- in a new SG in a new Paragraph.

Evaluation of a new semantic distance measure should then be useful at identifying words in the same POS, Paragraph and SG.

2 PREVIOUS WORK ON UPDATING THESAURI

There have been few attempts to expand the lexicon of *Roget's Thesaurus* thus far. Cassidy (2000) added manually a few hundred words to the 1911 edition of *Roget's Thesaurus*. Kennedy and Szpakowicz (2007) disambiguated hypernym instances in the 1987 *Roget's Thesaurus*. Both projects augmented *Roget's Thesaurus*, but did not offer insight into how to update the lexicon automatically.

Other related work includes mapping word senses between *Roget's Thesaurus*, *WordNet* and *LDOCE* (Procter 1978). The contexts where a word appears, whether it is words in the same Paragraph, *WordNet* synset or an *LDOCE* definition, are used to deduce which words are likely to be related (Kwong 1998a,b; Nastase and Szpakowicz 2001).

2.1 *Updating WordNet*

The automatic expansion of *WordNet*'s lexicon has been attempted several times. Snow *et al.* (2006) extracted thousands of new words from a corpus for possible inclusion in *WordNet* (though that expansion never materialised in practice due to its low accuracy). Many of the new terms were proper nouns found in a corpus by a machine learning system (Snow *et al.* 2005) which was used to discover IS-A relations using dependency paths generated by *MINIPAR* (Lin 1998b).

Pantel (2005) created semantic vectors for each word in *WordNet* by disambiguating contexts which appeared with different senses of a word. The building of semantic vectors is described in (Pantel 2003). *WordNet*'s hierarchy was used to propagate contexts where words may appear throughout the network. A word sense was then represented by contexts from its semantic vector not shared with its parents. Pantel did not attempt to place new words into the resource, only evaluated the method on existing words. This technique was only examined for nouns. It presumably applied to verbs as well, but could not be tried on adjectives or adverbs, for which there was no usable hypernym hierarchy.

A folksonomy is a Web service which allows users to annotate Web sites (among other things) with strings of their choice. One such folksonomy was *Delicious* where users categorised Web pages. HYPERNYM/HYPONYM relations can be extracted from folksonomies by identifying tags subsuming other tags. Zheng *et al.* (2008) describe how to use folksonomies to discover instances of hypernymy and so help put new words into *WordNet*.

Not directly applicable but relevant to our work is semi-automatic enhancement of *WordNet* with sentiment and affect information. Esuli and Sebastiani (2006) used machine learning to build *SentiWordNet* by labelling synsets in *WordNet* 2.0 as objective, positive or negative. In *WordNet Affect* (Strapparava and Valitutti 2004), synsets got one or more labels, often related to emotion. An initial set of words marked

with emotions was built manually. Next, those emotions were propagated to other synsets via *WordNet* relations. This work was based on *WordNet Domains* (Magnini and Cavagliá 2000), a framework which allows a user to augment *WordNet* by adding domain labels to synsets. No new words were added, but these projects highlight some of the more successful experiments with enhancing *WordNet*.

There is a reasonable amount of work on mining hypernym relations from text, which could then be used to update *WordNet*. This includes using set patterns (Hearst 1992; Sombatsrisomboon *et al.* 2003) or discovering new patterns using a few seed sets of hypernyms (Morin and Jacquemin 1999). Languages other than English for which hypernym mining has been attempted include Swedish (Rydin 2002), Dutch (Sang 2007) and Japanese (Shinzato and Torisawa 2004). There also has been research on hierarchically related verbs (Girju *et al.* 2003, 2006).

2.2 *Wordnets in other languages*

There has been much work on building wordnets for languages other than English, loosely coordinated by the Global Wordnet Association.[4] One strategy is to take the Princeton *WordNet* (Fellbaum 1998) as a starting point. That was the mode of operation in the *EuroWordNet* project (Vossen 1998), an early initiative meant to build wordnets for several European languages. One of its offshoots is *BalkaNet*.[5]

The other wordnet-building strategy is to avoid the influence of Princeton *WordNet*. Polish *WordNet* (Piasecki *et al.* 2009) is one such resource built from scratch. Its development was supported, among others, by *WordNet Weaver*, a tool which helps increase the vocabulary of a wordnet. The tool implements a two-phase algorithm. Phase I identifies a network vicinity in which to place a new word, while phase II connects possible candidate synsets. Phase II is semi-automatic: it is the linguists who decide what additions are ultimately made to the growing Polish *WordNet*.

[4] See `http://globalwordnet.org/wordnets-in-the-world/` for an up-to-date list of available wordnets.

[5] "The Balkan WordNet aims at the development of a multilingual lexical database comprising of individual WordNets for the Balkan languages." (`http://www.dblab.upatras.gr/balkanet/`)

Lemnitzer *et al.* (2008) discuss adding semantic relationships between nouns and verbs to *GermaNet*, a German wordnet. Those were verb–object relationships believed to be useful in applications such as text summarisation or anaphora resolution. Sagot and Fišer (2011) present an automatic, language-independent method (tested on Slovene and French) of extending a wordnet by "recycling" freely available bilingual resources such as machine-readable dictionaries and on-line encyclopaedias.

3 MEASURING SEMANTIC RELATEDNESS

Distributional measures of semantic relatedness (MSRs) use a word's context to help determine its meaning. Words which frequently appear in similar contexts are assumed to have similar meaning. Such MSRs usually re-weight contexts by considering some measure of their importance, usually the association between a context and the terms it contains. One of the most successful measures is Pointwise Mutual Information (PMI). PMI increases the weight of contexts where a word appears regularly but other words do not, and decreases the weight of contexts where many words may appear. Essentially, it is unsupervised feature weighting.

Kennedy and Szpakowicz (2011, 2012) discussed introducing supervision into the process of context re-weighting. Their method identifies contexts where pairs of words known to be semantically related frequently appear, and then uses a measure of association to re-weight these contexts by how often they contain closely related words. The method, very general, can work with any thesaurus as a source of known synonym pairs and with measures of association other than PMI. Here, this measure will help update *Roget's Thesaurus*. This section describes in general how this method is applied.

3.1 *Building a word–context matrix for semantic relatedness*

We used Wikipedia[6] as a source of data and parsed it with *MINIPAR* (Lin 1998a). The choice of dependency triples instead of all neighbouring words favours contexts which most directly affect a word's meaning. Examples of triples are $\langle time, mod, unlimited \rangle$ and

[6] A dump of August 2010.

$\langle time, conj, motion \rangle$: "time" appears in contexts with the modifier "unlimited" and in a conjunction with "motion". Some 900 million dependency triples generated by parsing Wikipedia took up ≈20GB.

Three matrices were built, one each for nouns, verbs and adjectives/adverbs.[7] For each word–relation–word triple $\langle w_1, r, w_2 \rangle$ we generated two word–context pairs $(w_1, \langle r, w_2 \rangle)$ and $(w_2, \langle w_1, r \rangle)$. Words w_1 and w_2 could be of any part of speech. All relations r were considered, with the direction of r retained. When w_1 or w_2 was an individual term, it had to be a noun, verb, adjective or adverb, written in lower case (*MINIPAR* only leaves proper nouns capitalised).

With these constraints we used all of the Wikipedia dump when building the matrices for verbs and adjectives/adverbs, but only 50% for nouns. This limit was chosen both because it was the most data which could be held in a system with 4GB of RAM and because the leftover data could be used in later evaluation.

Very infrequent words and contexts tend to be unreliable, and often appear because of spelling errors. We established thresholds for how often a word or context needs to appear. We measured the quality of synonyms generated for a set of randomly selected words which appear with different frequencies in the matrix. Next, in a series of straightforward experiments, we selected a cutoff after which the quality of the synonyms does not appear to improve: 35 for nouns and for adjectives, 10 for verbs. Also, an entry must appear in a context at least twice for the context to count. Table 1 shows the counts of words and contexts in each matrix before and after the cutoff. Non-zero entries are cells with positive values. While the reduction of the matrix dimensionally was quite large, the decrease in the number of non-zero entries was very small. So, we lost little information, but created a much denser and more informative matrix.

3.2 *Measures of semantic relatedness*

We explored two complementary methods of re-weighting the word-context matrix. An unsupervised method measures association between words and contexts; a supervised method uses known pairs of synonyms in *Roget's Thesaurus* to determine which contexts have a

[7] We have decided to work with *MINIPAR*'s labelling system, which does not distinguish between adjectives and adverbs.

Table 1: Counts of the rows, columns and non-zero entries for each matrix

POS	Matrix	Words	Contexts	Non-zero entries	% non-zero
Noun (≥ 35)	Full	359 380	2 463 001	30 994 968	0.0035%
	Cutoff	43 834	1 050 178	28 296 890	0.0615%
	(*% of full*)	(*12.2%*)	(*42.6%*)	(*91.3%*)	
Verb (≥ 10)	Full	9 294	2 892 002	26 716 709	0.0994%
	Cutoff	7141	1 423 665	25 239 485	0.2483%
	(*% of full*)	(*76.8%*)	(*49.3%*)	(*94.5%*)	
Adj/Adv (≥ 35)	Full	104 074	817 921	9 116 741	0.0107%
	Cutoff	17 160	360 436	8 379 637	0.1355%
	(*% of full*)	(*16.5%*)	(*44.1%*)	(*91.9%*)	

higher tendency to contain pairs of known synonyms (Kennedy and Szpakowicz 2011, 2012). Supervision can be conducted on each individual context, or on groups of contexts with a syntactic relation in common. It was found that supervision at the context level worked best for nouns and verbs, while grouping contexts by relation worked best for adjectives (Kennedy and Szpakowicz 2012).

Both supervised and unsupervised methods employ measures of association; Kennedy and Szpakowicz (2012) found that in all cases PMI was the most successful. These two kinds of methods can actually be complementary. It is possible to use the supervised method of matrix re-weighting and then apply the unsupervised method on top of it. This was generally found to yield the best results; so this is how we report the results.

To evaluate this work, we created a random set of 1000 nouns, 600 verbs and 600 adjectives and generated lists of neighbouring words for each of them.[8] Those words were left out of the training process. We then measured the precision – how many neighbouring words appeared in the same SG, Paragraph or POS – in the 1987 *Roget's Thesaurus*. Precision was measured at several recall points: the top 1, 5, 10, 20, 50 and 100 words retrieved from the 1987 *Thesaurus*.

Table 2 shows the results for the unsupervised baseline, using PMI weighting and the results for the combined supervised methods using synonyms from either the 1911 or the 1987 version of *Roget's*

[8] There were not enough adverbs to construct such a set. Adverbs will be left for future work.

Year	POS	Group	Top 1	Top 5	Top 10	Top 20	Top 50	Top 100
–	N.	SG	0.358	0.236	0.179	0.130	0.084	0.059
		Para	0.560	0.469	0.412	0.352	0.279	0.230
		POS	0.645	0.579	0.537	0.490	0.423	0.374
	V.	SG	0.302	0.206	0.162	0.126	0.086	0.065
		Para	0.513	0.445	0.407	0.358	0.304	0.264
		POS	0.582	0.526	0.487	0.444	0.396	0.357
	Adj.	SG	0.345	0.206	0.156	0.115	0.069	0.046
		Para	0.562	0.417	0.363	0.304	0.231	0.185
		POS	0.600	0.480	0.431	0.368	0.295	0.247
1911	N.	SG	0.358	*0.225*	*0.175*	0.132	0.084	0.058
		Para	0.568	0.472	0.418	**0.361**	**0.286**	**0.234**
		POS	0.659	**0.588**	**0.548**	**0.501**	**0.431**	**0.382**
	V.	SG	0.310	0.207	0.163	0.124	0.086	0.064
		Para	**0.550**	0.456	0.414	0.362	0.307	**0.268**
		POS	0.605	0.533	**0.500**	**0.455**	**0.401**	**0.362**
	Adj.	SG	0.343	**0.209**	0.157	0.114	0.069	0.046
		Para	0.563	0.422	0.365	0.304	**0.232**	0.184
		POS	0.602	0.484	0.431	0.368	0.296	0.247
1987	N.	SG	0.359	*0.229*	0.177	**0.134**	0.085	0.059
		Para	0.564	0.471	**0.419**	**0.365**	**0.285**	**0.234**
		POS	0.651	0.584	**0.549**	**0.501**	**0.430**	**0.381**
	V.	SG	0.308	0.211	0.167	0.127	0.087	0.064
		Para	0.525	**0.457**	**0.417**	0.362	0.305	**0.266**
		POS	0.588	**0.537**	**0.499**	**0.453**	0.399	**0.360**
	Adj.	SG	0.343	0.208	0.158	0.115	0.069	0.046
		Para	0.565	**0.421**	0.365	0.304	0.232	0.184
		POS	0.603	0.483	0.431	0.367	0.296	0.247

Table 2: Evaluation results for the combined measure with PMI. Significant improvement over unsupervised PMI in **bold**, significantly worse results in *italics*

Thesaurus as training data. Statistically significant improvement over the baseline appears in bold, while significantly worse results are italicised; we applied Student's t-test. With a few small exceptions, we found that the supervised system performs better. The number of times the scores were better, unchanged, or worse can be found in Table 3. In general, we concluded that the combination of supervised and unsupervised context weighting created a superior MSR, better suited to

Table 3:
The number of statistically improved/unaffected/ decreased results for both sources of training data

Resource	Nouns	Verbs	Adjectives	All
1911 *Roget's*	8/8/2	6/12/0	2/16/0	16/36/2
1987 *Roget's*	9/8/1	7/11/0	1/17/0	17/36/1

updating *Roget's Thesaurus* than the unsupervised method alone. We used the supervised method of generating lists of related words when adding new terms to the *Thesaurus*.

4 PLACING NEW WORDS IN *ROGET"S THESAURUS*

In this section, we evaluate a variety of systems for adding new words to *Roget's Thesaurus*. The baseline method places a word in the same POS, Paragraph and Semicolon Group as its closest neighbour in the *Thesaurus*. We improve on this baseline using multiple words to deduce a better location or better locations.

4.1 *Methods of adding new words*

We took advantage of the hierarchy of *Roget's Thesaurus* to select the best place to add words. We found first the POS, then the Paragraph, then the SG.[9] We refer to the word to be added to *Roget's Thesaurus* as the *target* word. A word already in the *Thesaurus* may be an *anchor*, acting as a "magnet" for a given target. For every target word t, we generated a list of nearest neighbours $NN(t)$, along with similarity scores, and identified anchors using $NN(t)$.

We experimented with three methods, evaluated against the following baseline: the target t is placed in the same POS, Paragraph and SG as w_i, where w_i is the first word in $NN(t)$ found in *Roget's Thesaurus*. Since w_i may be polysemous, t can go into multiple locations in *Roget's Thesaurus*. Often w_i will be w_1 if the first neighbour of t is found in the *Thesaurus*. For the values in Table 4, this baseline has been calculated using the MSRs built with combined weighting, trained with the 1911 or the 1987 *Thesaurus*. The results show one number for the count of POSs, Paragraphs and SGs where the target t was placed and the precision of placing the word into the POSs, Paragraphs and SGs.

[9] Identifying the POS effectively gives us the correct Head as well.

The first method is to apply a nearest-neighbour model. X nearest neighbours from $NN(t)$ are identified for each target word t. If W of these X words appear in the same *Roget's grouping*, the target word is placed there. It is a weakness that this method considers – somewhat unrealistically – the same number of neighbours for every target word.

In the second method, scores replace rank. Words with scores of Y or higher are identified. If W of them are in the same *Roget's grouping*, the target word is placed there. This allows for varying numbers of neighbours, but similarity scores partially depend on the target word, so the same score between two different word pairs may indicate different degrees of similarity. A very frequent word which appears in many contexts may have more highly related neighbours than a word which appears in few contexts. Such a frequent word may thus have inordinately many synonyms.

The third method considers relative scores. It assumes that the first similar word w_1 is very closely related to t, then takes all synonyms within Z% of the similarity score for w_1. This means that if w_i has a score of within Z% of w_1, then it can be used as an anchor of t for determining the correct *Roget's grouping*. Once again, if W of these words in the same *Roget's grouping* have a relative score of Z% or higher, then the target word can be placed there as well.

We also considered how to optimise the measures. In placing words into a *Roget's grouping*, the method has two parameters to optimise, W and one of X, Y or Z. One possibility is to base F-measure on the precision with which words are placed in *Roget's Thesaurus* and recall on the number of words from the test set which could actually be placed. Another possibility of counting recall would be to identify the number of places where a word appears in the *Thesaurus* and see in how many of them it *was* placed. This measure has some problems.

For one, rare senses are not well represented by the vectors in the word–context matrix, so synonyms for only the most dominant senses will be found. Also, an even balance of precision and recall is not appropriate for this task. Adding incorrect words could be quite detrimental, so we assume that identifying the POS must weight precision more highly than recall. We set a 0.33 ratio of recall to precision (an F0.33 measure rather than F1). Once the POS has been identified, the Paragraph and SG will be identified using the F1 measure. The

Table 4: Baseline for identifying the POS of a word on the tuning and test data

Year	POS	Data	Words	P	R	F0.33
1987	Noun	*Tuning*	*1000*	*0.281*	*0.486*	*0.293*
		Test	1000	0.295	0.487	0.307
	Verb	*Tuning*	*600*	*0.204*	*0.468*	*0.216*
		Test	600	0.245	0.455	0.257
	Adjective	*Tuning*	*600*	*0.250*	*0.460*	*0.262*
		Test	600	0.232	0.435	0.244
1911	Noun	*Tuning*	*817*	*0.232*	*0.296*	*0.237*
		Test	840	0.267	0.344	0.273
	Verb	*Tuning*	*542*	*0.167*	*0.271*	*0.174*
		Test	538	0.196	0.297	0.203
	Adjective	*Tuning*	*489*	*0.246*	*0.288*	*0.249*
		Test	497	0.201	0.262	0.206

choice of F0.33 is somewhat arbitrary, but favouring precision over recall should mostly bring advantages. A high-precision system is, in theory, more likely to place words in the correct *Roget's grouping* at the cost of lower recall. Any method of adding new words to *Roget's Thesaurus*, however, could be run iteratively and thus make up for the lower recall. Rather than attempting to add a lot of words in one pass, our method will add fewer words in each of multiple passes.

When using this method to actually add new words, sometimes it is necessary to create new Paragraphs or SGs. If a POS is identified but no Paragraph, then a new Paragraph will be created. Likewise, if a Paragraph but not an SG can be identified, then the word is placed in a new SG in the selected Paragraph.

The methods were tuned on the same dataset as that used to evaluate the MSR in Section 3. For evaluation, we constructed a test set equal in size to the tuning set. We evaluated all methods on the task of identifying the correct POS to place a target word t. The best method is then applied to the task of placing a word in the appropriate Paragraph and SG.

4.2 *Baseline*

Table 4 shows the results of the baseline experiments, measured for the 1911 and 1987 versions of *Roget's Thesaurus*. The former did not contain all the words for evaluation that the latter did – hence

		1987		1911	
Parameter	POS	$X/Y/Z$	W-POS	$X/Y/Z$	W-POS
X	Noun	26	10	10	4
	Verb	22	7	6	3
	Adjective	19	6	8	3
Y	Noun	.08	15	.07	14
	Verb	.09	9	.13	2
	Adjective	.13	3	.10	4
Z	Noun	.82	4	.93	2
	Verb	.89	3	.98	2
	Adjective	.82	3	.91	2

Table 5: Optimal values for parameters X (the number of nearest neighbours), Y (the minimal relatedness score) and Z (the relative score)

Year	POS	Data	Words	P	R	F0.33
1987	Noun	*Tuning*	*1000*	*0.746*	*0.267*	*0.633*
		Test	1000	0.758	0.262	0.637
	Verb	*Tuning*	*600*	*0.565*	*0.285*	*0.514*
		Test	600	0.536	0.252	0.482
	Adjective	*Tuning*	*600*	*0.658*	*0.273*	*0.577*
		Test	600	0.590	0.233	0.512
1911	Noun	*Tuning*	*817*	*0.613*	*0.171*	*0.488*
		Test	840	0.659	0.182	0.522
	Verb	*Tuning*	*542*	*0.484*	*0.131*	*0.381*
		Test	538	0.471	0.097	0.340
	Adjective	*Tuning*	*489*	*0.571*	*0.184*	*0.472*
		Test	497	0.503	0.141	0.400

Table 6: Precision, Recall and F0.33-measure when optimising for X, the number of nearest neighbours

the differences in word counts. The results show a small advantage of adding words to the 1987 *Thesaurus* over the 1911 version.

4.3 *Tuning parameters for adding new words*

Table 5 shows the parameters, optimised for F0.33, for the three non-baseline methods. Tables 6–8 present the results on the tuning and test data.

When optimising for the X nearest neighbours (Table 6), the results show a large improvement over the baseline (Table 4). The results for nouns were actually better on the test dataset than on tuning data,

Table 7: Precision, Recall and F0.33-measure when optimising for Y, the minimal relatedness score

Year	POS	Data	Words	P	R	F0.33
1987	Noun	*Tuning*	*1000*	*0.596*	*0.182*	*0.486*
		Test	1000	0.507	0.160	0.417
	Verb	*Tuning*	*600*	*0.477*	*0.078*	*0.316*
		Test	600	0.573	0.062	0.313
	Adjective	*Tuning*	*600*	*0.529*	*0.122*	*0.396*
		Test	600	0.421	0.103	0.322
1911	Noun	*Tuning*	*817*	*0.420*	*0.120*	*0.336*
		Test	840	0.367	0.110	0.297
	Verb	*Tuning*	*542*	*0.211*	*0.096*	*0.189*
		Test	538	0.234	0.063	0.184
	Adjective	*Tuning*	*489*	*0.480*	*0.084*	*0.326*
		Test	497	0.274	0.066	0.209

but somewhat worse for verbs and adjectives. As with the baseline, the results were better for the 1987 *Roget's Thesaurus* than the 1911 version. Generally about one third to half of the words found in the top X needed to be present in the same *Roget's grouping* in order to accurately select the correct grouping.

Table 7 shows optimising word placement with scores Y or higher. The optimal scores were noticeably lower than when we optimised for X nearest neighbours (Table 6). The minimum score Y appeared to be lower for nouns than for verbs or adjectives, though more words were required in order to identify the *Roget's grouping* positively. This method is not as successful as simply selecting the X nearest neighbours. For verbs added to the 1911 *Roget's Thesaurus*, there was actually no improvement over the baseline (Table 4). This is the least successful method of the three.

Table 8 reports on optimising for the relative score Z. We found that most neighbouring words had to be within 80–90% of the closest neighbour in terms of score. This improved the results noticeably over a simple selection of a hard score cut-off (Table 7). Nonetheless, we did not improve on simply taking the X nearest neighbours (Table 6). For determining relatedness, it would appear, rank is often a feature more important than score. With this in mind, we applied the nearest-neighbour function using X to find the best parameters for identifying the POS, Paragraph and SG. The parameter W shown in Table 5 was

Year	POS	Data	Words	P	R	F0.33
1987	Noun	*Tuning*	*1000*	*0.643*	*0.190*	*0.519*
		Test	1000	0.595	0.215	0.506
	Verb	*Tuning*	*600*	*0.468*	*0.147*	*0.384*
		Test	600	0.492	0.163	0.410
	Adjective	*Tuning*	*600*	*0.512*	*0.215*	*0.450*
		Test	600	0.463	0.200	0.409
1911	Noun	*Tuning*	*817*	*0.468*	*0.200*	*0.413*
		Test	840	0.542	0.219	0.473
	Verb	*Tuning*	*542*	*0.438*	*0.118*	*0.344*
		Test	538	0.389	0.091	0.293
	Adjective	*Tuning*	*489*	*0.478*	*0.145*	*0.389*
		Test	497	0.434	0.129	0.351

Table 8:
Precision, Recall and F0.33-mesure when optimising for Z, the relative score

Year	POS	X	W-POS	W-Para	W-SG
1987	Noun	26	10	5	2
	Verb	22	7	4	3
	Adjective	19	6	4	2
1911	Noun	10	4	3	3
	Verb	6	3	2	2
	Adjective	8	3	2	2

Table 9:
Optimal parameters for X (the number of nearest neighbours) and W (neighbours needed to insert a word into a *Roget's grouping*) at the POS, Paragraph and SG levels

for the POS level. We have three versions, W-POS, W-Para and W-SG for the POS, Paragraph and SG respectively.

Table 9 shows the optimal values of X, W-POS, W-Para and W-SG. The same value of X was used for identifying groupings at the POS, Paragraph and SG levels. There is a bit of variance in the measures. The values of W-POS, W-Para and W-SG decrease as the groupings become smaller. To identify the correct SG, only 2 or 3 words were used. For the 1911 *Roget's Thesaurus*, the same number of words were used to identify the Paragraph as the SG. More words could be used to identify the POS for the 1987 *Thesaurus* than for the 1911 version.

Tables 10 and 11 show the precision, recall and F1 measure at the POS, Paragraph and SG level for the 1987 and 1911 *Thesauri*. The results show clearly that the F1 measure is highest when identifying the Paragraph level; this is largely because the POS level is optimised for the F0.33 measure. Once again, the scores for the 1987 version

Table 10: Identifying best POS, Paragraph and SG using optimised values for X, W-POS, W-Para and W-SG, using the F1 measure for evaluation on the **1987** *Roget's Thesaurus*

	Data	RG	P	R	F1
Noun	*Tuning*	*POS*	*306/410 (0.746)*	*267/1000 (0.267)*	*0.393*
	Tuning	*Para*	*225/402 (0.560)*	*189/267 (0.708)*	*0.625*
	Tuning	*SG*	*104/664 (0.157)*	*92/189 (0.487)*	*0.237*
	Test	POS	304/401 (0.758)	262/1000 (0.262)	0.389
	Test	Para	234/416 (0.562)	196/262 (0.748)	0.642
	Test	SG	101/659 (0.153)	93/196 (0.474)	0.232
Verb	*Tuning*	*POS*	*227/402 (0.565)*	*171/600 (0.285)*	*0.379*
	Tuning	*Para*	*186/413 (0.450)*	*137/171 (0.801)*	*0.577*
	Tuning	*SG*	*34/129 (0.264)*	*32/137 (0.234)*	*0.248*
	Test	POS	185/345 (0.536)	151/600 (0.252)	0.343
	Test	Para	148/339 (0.437)	114/151 (0.755)	0.553
	Test	SG	18/103 (0.175)	17/114 (0.149)	0.161
Adj	*Tuning*	*POS*	*227/345 (0.658)*	*164/600 (0.273)*	*0.386*
	Tuning	*Para*	*182/312 (0.583)*	*136/164 (0.829)*	*0.685*
	Tuning	*SG*	*75/381 (0.197)*	*63/136 (0.463)*	*0.276*
	Test	POS	193/327 (0.590)	140/600 (0.233)	0.334
	Test	Para	152/294 (0.517)	116/140 (0.829)	0.637
	Test	SG	59/351 (0.168)	51/116 (0.440)	0.243

tend to be better than those for the 1911 version. Most of the time it is possible to identify the correct POS with at least 40% accuracy. The recall for the 1987 *Thesaurus* was 0.233 or higher at the POS level. This is important, because it indicates how many new word additions to the *Thesaurus* can be expected. For the 1911 *Thesaurus*, the results tend to be much lower, with scores from 0.097 to 0.182 on the test set. The number for verbs is very low; for nouns and adjectives it is better, but still lower than the corresponding results for the 1987 thesaurus.

4.4 *Adding words to the Thesaurus*

We now show how the method described in Section 4.3 adds words to *Roget's Thesaurus*. In practice, a few small modifications were needed. First, we only let a word be placed in a POS if it was not already present in either that POS or in another POS within the same Head Group. This reduced the possibility of entering antonyms, which may be distributionally similar, into the same POS. Within each POS, we let a word be placed only in one Paragraph. We also did not allow

	Data	RG	P	R	F1
Noun	*Tuning*	*POS*	*157/256 (0.613)*	*140/817 (0.171)*	*0.268*
	Tuning	*Para*	*89/163 (0.546)*	*83/140 (0.593)*	*0.568*
	Tuning	*SG*	*31/62 (0.500)*	*29/83 (0.349)*	*0.411*
	Test	POS	162/246 (0.659)	153/840 (0.182)	0.285
	Test	Para	83/155 (0.535)	78/153 (0.510)	0.522
	Test	SG	29/55 (0.527)	28/78 (0.359)	0.427
Verb	*Tuning*	*POS*	*76/157 (0.484)*	*71/542 (0.131)*	*0.206*
	Tuning	*Para*	*55/136 (0.404)*	*53/71 (0.746)*	*0.525*
	Tuning	*SG*	*24/86 (0.279)*	*24/53 (0.453)*	*0.345*
	Test	POS	57/121 (0.471)	52/538 (0.097)	0.160
	Test	Para	39/112 (0.348)	35/52 (0.673)	0.459
	Test	SG	22/76 (0.289)	19/35 (0.543)	0.378
Adj	*Tuning*	*POS*	*109/191 (0.571)*	*90/489 (0.184)*	*0.278*
	Tuning	*Para*	*80/188 (0.426)*	*71/90 (0.789)*	*0.553*
	Tuning	*SG*	*23/107 (0.215)*	*22/71 (0.310)*	*0.254*
	Test	POS	79/157 (0.503)	70/497 (0.141)	0.220
	Test	Para	46/148 (0.311)	42/70 (0.600)	0.409
	Test	SG	14/91 (0.154)	13/42 (0.310)	0.206

Table 11: Identifying best POS, Paragraph and SG using optimised values for X, W-POS, W-Para and W-SG, using the F1 measure for evaluation on the **1911** *Roget's Thesaurus*

adding the same word to multiple SGs within the same Paragraph or indeed to multiple Paragraphs in the same POS.

Once a new word has been added to *Roget's Thesaurus*, it can be used as an anchor to help add subsequent words. We built two updated versions of each *Thesaurus*, one with a single pass to update the *Thesaurus*, another with five updating passes. We considered each word in each matrix, excluding stop words,[10] to be a target and generated a list of the nearest 100 neighbours for each of these words.[11] It was from these lists that we attempted to add new words to the *Thesaurus*.

Several measures are of interest when adding new words to the *Thesaurus*. The first is the number of times a target word has sufficient X and W values to be placed in *Roget's Thesaurus*, regardless of whether it was already present. The second measure is the total num-

[10] We applied a 980-element union of five stop lists first used in Jarmasz (2003): Oracle 8 ConText, SMART, Hyperwave, a list from the University of Kansas and a list from Ohio State University.

[11] Only the top X of those 100 helped identify the best place for a new word.

ber of words added to the *Thesaurus*. The third measure is the number of unique words added. These two are likely to be similar since most often a target word is only added to a single location in the *Thesaurus*. The fourth measure counts new words whose derivational form already exists in the *Thesaurus*. The fifth measure counts new words which have no derivationally related words in the *Thesaurus*. The last measure is the number of Heads where a new word was added. The results for all five passes can be seen in Table 12.

In addition to the five passes of adding new words, we experimented with random addition. All process parameters are the same, up to the point when our system determines a location where it believes a word belongs. Before checking whether that word already appears at this location, it is swapped for a random word. The counts appear in Table 13. Since the random word is selected after a location has been decided, it is very rare for this word already to be in that Head Group. As a result, the number of attempted placements is very close to the total number of words added, much closer than for the counts from Table 12.

Ultimately three updated version each of the 1911 and 1987 versions of the *Thesaurus* were created, those updated with one pass, five passes and one random pass – X1, X5 and R in Table 14. The updated versions are referred to as 1911X1, 1911X5, 1911R, 1987X1, 1987X5 and 1987R. The new thesauri have been evaluated manually (Section 5) and through selected NLP applications (Section 6).

Another statistic to consider is the total number of words, SGs and Paragraphs added to each version of *Roget's Thesaurus*, shown in Table 14. Overall, some 5500 new words were added to 1911X5 and 9600 to 1987X5. In the 1911 *Thesaurus*, approximately two thirds of the new words were placed in a new SG, while about a quarter were added to a new Paragraph. For the 1987 *Thesaurus*, a little under half of the new words were placed in new SGs, while around one fifth were added to new Paragraphs.

5 MANUAL EVALUATION

To determine the quality of the additions reliably, one needs manual evaluation. In the next subsection, we describe several possibilities and explain how we chose our evaluation method.

P	Year	POS	Matches	Total Words	Unique Words	Derived Words	New Words	Heads Affected
1	1987	Nouns	6755	1510	1414	175	98	206
		Verbs	2870	893	735	52	45	129
		Adj	3053	858	713	15	10	183
	1911	Nouns	3888	1259	1193	148	68	274
		Verbs	1069	407	378	22	19	133
		Adj	1430	539	480	18	16	198
2	1987	Nouns	8388	774	742	37	14	139
		Verbs	4335	747	653	23	16	92
		Adj	4412	612	549	4	4	114
	1911	Nouns	5315	762	719	65	13	164
		Verbs	1530	247	238	14	14	71
		Adj	2083	287	262	6	5	95
3	1987	Nouns	9213	499	478	16	6	88
		Verbs	5303	600	543	16	14	61
		Adj	5275	532	463	7	2	80
	1911	Nouns	6109	549	520	35	11	100
		Verbs	1761	147	142	6	6	36
		Adj	2393	205	191	5	4	57
4	1987	Nouns	9767	384	378	11	2	60
		Verbs	6068	523	496	11	9	49
		Adj	5926	451	404	6	6	55
	1911	Nouns	6652	417	395	20	5	76
		Verbs	1898	106	105	0	0	21
		Adj	2571	139	129	1	0	35
5	1987	Nouns	10210	330	324	12	2	49
		Verbs	6689	464	422	6	3	39
		Adj	6509	424	382	3	1	38
	1911	Nouns	7026	295	288	22	10	54
		Verbs	1979	76	74	0	0	14
		Adj	2710	119	115	1	0	22

Table 12: New words added after the 1^{st}, 2^{nd}, 3^{rd}, 4^{th} and 5^{th} pass (P)

Table 13: Random words added after one pass

Year	POS	Matches	Total Words	Unique Words	Derived Words	New Words	Heads Affected
1987	Nouns	6755	6189	5007	3923	3593	306
	Verbs	2870	2238	1366	734	715	186
	Adj	3053	2631	1670	1547	1488	278
1911	Nouns	3888	3718	3203	2736	2554	379
	Verbs	1069	946	759	468	465	195
	Adj	1430	1349	1051	952	926	276

Table 14: New Paragraphs, SGs and words in the updated versions of *Roget's Thesaurus*

Resource	New Paragraphs	New SGs	New Words
1911X1	633	1442	2209
1911X5	1851	3864	5566
1911R	1477	3803	6018
1987X1	653	1356	3261
1987X5	2063	4466	9601
1987R	1672	3731	11058

5.1 *Methods considered*

The first evaluation method would test how well people can identify newly added words. Given a set of Paragraphs from *Roget's Thesaurus*, the annotator would be asked to identify which words she thought were added automatically and which were originally in the *Thesaurus*. The percentage of times the annotator correctly identifies newly added words can be used to evaluate the additions. If a word already in the *Thesaurus* were as likely to be picked as one newly added, then the additions would be indistinguishable – an ideal outcome. We could also perform a "placebo test": the annotator gets a Paragraph where no words have been added, and decides whether to remove any words at all. A drawback is that the annotator may be more likely to select words whose meaning she does not know, especially in the 1911 *Thesaurus*, where there are many outdated words. Even the 1987 version has many words infrequently used today.

The second method of manual evaluation we considered was to ask the annotator to assign a new word to the correct location in the *Thesaurus*. A weighted edit-distance score could then tell how many steps the system's placement is from that location. We would also mea-

Score	*Roget's* Paragraph
(word fits in this SG)	*Head 25: Agreement, noun* fitness, aptness; relevancy; pertinence, pertinencey; sortance; case in point; **aptitude, coaptation, propriety, applicability, admissibility, commensurability, *compatibility*;** cognation.

Figure 3: Example of the annotator task for adding a word to a Paragraph

sure how often the annotator needed to create a new Paragraph or SG for the word, and how many SGs and Paragraphs were automatically created but should not have been. Such a method would be labour-intensive: the annotator would need to read an entire Head before deciding how far a word is from its correct location. Larger Heads, where most new words are added, could contain thousands of words. Identifying whether there is an SG more appropriate for a given word could also take a fair bit of effort. It might not be feasible to annotate enough data to perform a meaningful evaluation.

The strategy we finally adopted combines elements of the two preceding methods. The first step of this evaluation exercise is to decide whether new words added to an existing SG or a new SG in an existing Paragraph are in the correct location. The annotator is given the name of the Head, the part of speech and the text of the Paragraph where the word has been added. The new term is specially highlighted, and other terms in its SG are in bold. The annotator is asked to decide whether the new word is in the correct SG, wrong SG but correct Paragraph, wrong Paragraph but correct Head, or incorrect Head. Figure 3 shows a sample question.

The second evaluation step determines whether a word added to a new Paragraph is in the correct Head. As context, we provide the first word in every Paragraph in the same POS. It is too onerous to determine precisely in which SG or Paragraph a new word would belong, because some POSs are very large. Instead, we only ask whether the word is in the correct Head. A sample question appears in Figure 4.

Figure 4: Example of the annotator task for adding a word to a POS

Score	*Roget's* Paragraph
(closely related)	*Head 25: Agreement, noun* agreement.. / conformity.. / fitness.. / adaption.. / ***consent;***

We manually evaluated only the additions to the 1911 *Roget's Thesaurus*. As Paragraph size, we allowed at most 250 characters, thus limiting the number of words the annotators had to look at. The evaluation was completed by the first author and four volunteers. We chose enough samples to guarantee a 5% confidence interval at a 95% confidence level.[12] We also included a high baseline and a low baseline: words already present in the *Thesaurus*[13] and words randomly added to it. There are enough samples from the baselines to guarantee a 5% confidence interval at a 95% confidence level if the samples from all three parts of speech are combined, though individually the confidence interval exceeds 5%.

Every new word in 1911X1 appears in 1911X5,[14] so a percentage of the samples needed to evaluate 1911X5 can be selected from the samples used to evaluate 1911X1. We thus must evaluate only a selection of the words from 1911X5 not present in 1911X1. We randomly selected words from the sample set for 1911X1 to make up the rest of the samples for the 1911X5 evaluation.

Random selection was made from each annotator's dataset: 40 tests for adding words to existing Paragraphs and 40 tests for adding words to new Paragraphs. These data points were added to each annotator's test sets so that there would be an overlap of 200 samples for each experiment, on which to calculate inter-annotator agreement. The positive examples are words already present in *Roget's Thesaurus*. The negative examples are words randomly placed in the *Thesaurus*.

[12] `http://www.macorr.com/sample-size-calculator.htm`

[13] They are referred to as "pre-existing" in Tables 15–16, in Figures 5–6 and in the discussion in Section 5.2

[14] We remind the reader that X1 and X5 denote updating with one pass and five passes respectively.

5.2 *Manual annotation results*

Tables 15 and 16 show the combined manual annotation results for new words added to existing Paragraphs and for new Paragraphs. A number of interesting observations can be taken from Table 15. The results are summarised in Figure 5. In the case of pre-existing examples, around 60% of the time the annotators could correctly determine when a word belonged in the SG in which it was found. The annotators agreed on the correct Head approximately 80–90% of the time. One reason why annotators might believe the words belonged in a different grouping was that many of the words were difficult to understand. A high number of words which the annotators could not label fell into the pre-existing category. For the randomly assigned words, 70–80% of the time the annotators correctly stated that those words did not belong in that Head. For nouns there were numerous cases when the annotators could not answer. It would appear that the meaning of words pre-existing in the *Thesaurus*, and of those randomly added, is harder to determine than the meaning of automatically added words.

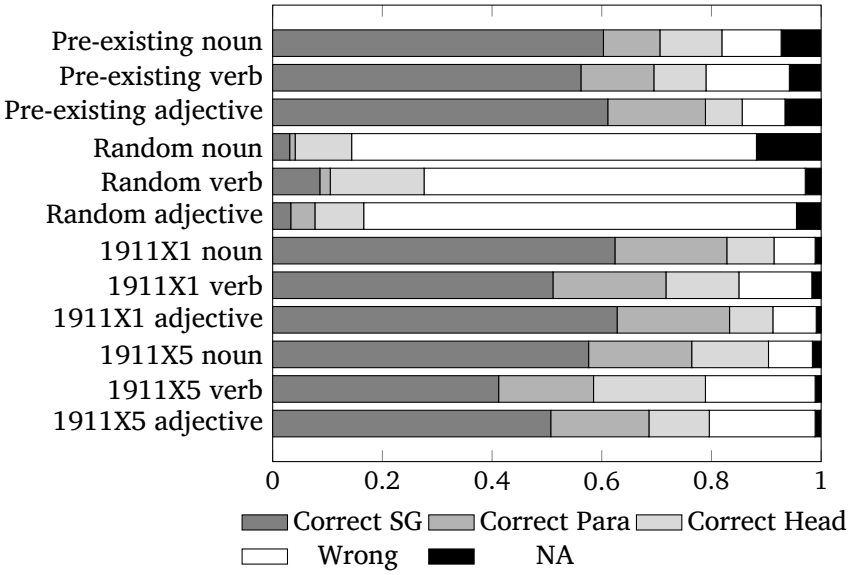
We now turn to the quality of additions. The distribution of 1911X1 scores in Table 15 is very close to that of the distribution for words pre-existing in *Roget's Thesaurus*. This suggests that after one pass the added words are nearly indistinguishable from those already in the *Thesaurus*. This is very good news: it confirms that our process of updating the lexicon has succeeded. The distribution of 1911X5 scores suggests that those additions were less reliable. The scores are worse than for 1911X1, but still much closer to the pre-existing baseline than the random baseline. Multiple passes increase the error, but not by much.

The results are a bit different when it comes to inserting words into new Paragraphs. These results are summarised in Figure 6. Once again the high and low baselines appeared to be fairly easy for the annotators, who usually got around 80% of the questions right. Also, a solid majority of the unknown words appeared in these two groups. The additions to 1911X1 showed high scores, too, comparable to the high baseline, sometimes even exceeding it slightly. It may be that for this baseline the annotators were unaware of the sense of some words, so they mistakenly labelled those words as incorrect.

Table 15: Manual evaluation results for words added to previously existing Paragraphs

Task	POS	Correct SG	Correct Para	Correct Head	Wrong Head	N/A
Pre-existing Words	Noun	117 (*.600*)	20 (*.103*)	22 (*.113*)	21 (*.108*)	15 (*.077*)
	Verb	59 (*.562*)	14 (*.133*)	10 (*.095*)	16 (*.152*)	6 (*.057*)
	Adj.	55 (*.611*)	16 (*.178*)	6 (*.067*)	7 (*.078*)	6 (*.067*)
Random Words	Noun	6 (*.031*)	2 (*.010*)	20 (*.103*)	144 (*.738*)	23 (*.118*)
	Verb	9 (*.086*)	2 (*.019*)	18 (*.171*)	73 (*.695*)	3 (*.029*)
	Adj.	3 (*.033*)	4 (*.044*)	8 (*.089*)	71 (*.789*)	4 (*.044*)
1911X1	Noun	159 (*.624*)	52 (*.204*)	22 (*.086*)	19 (*.075*)	3 (*.012*)
	Verb	92 (*.511*)	37 (*.206*)	24 (*.133*)	24 (*.133*)	3 (*.017*)
	Adj.	135 (*.628*)	44 (*.205*)	17 (*.079*)	17 (*.079*)	2 (*.009*)
1911X5	Noun	181 (*.576*)	59 (*.188*)	44 (*.140*)	25 (*.080*)	5 (*.016*)
	Verb	107 (*.412*)	45 (*.173*)	53 (*.204*)	52 (*.200*)	3 (*.012*)
	Adj.	147 (*.507*)	52 (*.179*)	32 (*.110*)	56 (*.193*)	3 (*.010*)

Figure 5: Evaluation on words added to previously existing Paragraphs in the 1911 *Roget's Thesaurus*



Task	POS	Correct Head	Wrong Head	N/A
Pre-existing Words	Noun	158 (*.810)*	33 (*.169)*	4 (*.021)*
	Verb	87 (*.829)*	17 (*.162)*	1 (*.010)*
	Adj	75 (*.833)*	14 (*.156)*	1 (*.011)*
Random Words	Noun	18 (*.092)*	151 (*.774)*	26 (*.133)*
	Verb	17 (*.162)*	83 (*.790)*	5 (*.048)*
	Adj	13 (*.144)*	74 (*.822)*	3 (*.033)*
1911X1	Noun	189 (*.859)*	27 (*.123)*	4 (*.018)*
	Verb	50 (*.833)*	10 (*.167)*	0 (*.000)*
	Adj	48 (*.873)*	7 (*.127)*	0 (*.000)*
1911X5	Noun	207 (*.674)*	94 (*.306)*	6 (*.020)*
	Verb	64 (*.533)*	55 (*.458)*	1 (*.008)*
	Adj	61 (*.616)*	37 (*.374)*	1 (*.010)*

Table 16: Manual evaluation results for words added to new Paragraphs

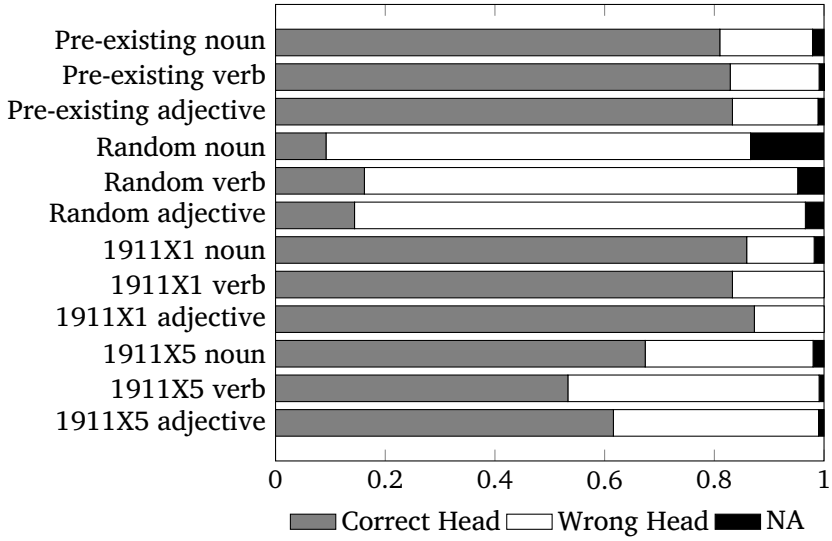


Figure 6: Evaluation on words added to new Paragraphs in the 1911 *Roget's Thesaurus*

The 1911X5 results – multi-pass update – clearly fall a fair distance from the scores for 1911X1. It would appear that multiple passes introduce considerable error into the *Thesaurus*, when words are placed into new Paragraphs. This is in stark contrast to the result of adding words to existing Paragraphs, when the drop in scores between 1911X1 and 1911X5 was relatively small.

5.3 *Inter-annotator agreement*

Each annotator was given 200 examples which reoccurred between the annotations. Inter-annotator agreement was measured on these overlaps, using Krippendorff's α (Krippendorff 2004), a measure designed to work with various kinds of data, including nominal, ordinal and interval annotations. We used ordinal in our experiments. The value of α was calculated for adding words both to existing Paragraphs and to new Paragraphs. When adding words to an existing Paragraph, we obtained a score of $\alpha = 0.340$; when adding words to new Paragraphs, the score was $\alpha = 0.358$. Such scores are often considered a "fair" amount of agreement (Landis and Koch 1977).

6 AUTOMATIC EVALUATION

We now examine how the various versions of *Roget's Thesaurus*, as well as *WordNet* 3.0, perform on several NLP applications. The problems we selected are designed to evaluate *Roget's Thesaurus* on a diverse cross-section of NLP tasks: synonym identification, pseudo-word-sense disambiguation and SAT-style analogy problems. We use *WordNet* 3.0 and all available versions of the *Thesaurus*: 1911, 1911X1, 1911X5, 1911R, 1987, 1987X1, 1987X5 and 1987R. Although the updated versions of *Roget's Thesaurus* are larger than the original, and new words have been added with relatively high precision, there is no *a priori* guarantee that they will give higher scores on any NLP applications. Before we harness these resources into NLP applications, we will very briefly compare the structure of *Roget's Thesaurus* to that of *WordNet*.[15]

A major difference between *WordNet* and *Roget's Thesaurus* is that the former is built around a hypernym hierarchy of arbitrary depth. Words appear at all levels, rather than only at the bottom level, as in

[15] For a detailed presentation of *WordNet*, see (Fellbaum 1998).

Roget's Thesaurus. Words are grouped into synsets. Synsets are similar to SGs in the *Thesaurus*, but are often smaller and contain only close synonyms. Synsets are linked by a variety of explicitly named semantic relations, while in the *Thesaurus* the SGs in a Paragraph are loosely related by a variety of possible implicit relations.

6.1 *Synonym identification*

Synonym identification is a means of evaluating the quality of newly added words in *Roget's Thesaurus*. In this problem one is given a term q and seeks its best synonym s in a set of words C. The system from Jarmasz and Szpakowicz (2003b, 2004) identifies synonyms using the *Thesaurus* as the lexical resource. This method relies on a simple function which counts the number of edges in the *Thesaurus* between q and words in C. In Equation 1, 18 is the highest possible distance in the *Thesaurus*, so the closest words have the highest scores (*edgesBetween* simply counts the edges). We treat a word X as a lexeme: a set of word senses $x \in X$.

$$edgeScore(X, Y) = \max_{x \in X,\, y \in Y} [18 - edgesBetween(x, y)] \tag{1}$$

The best synonym is selected in two steps. First, we find a set of terms $B \subseteq C$ with the maximum relatedness between q and each word sense $x \in C$ (Equation 2).

$$B = \left\{ x \mid \operatorname*{argmax}_{x \in C} edgeScore(x, q) \right\} \tag{2}$$

Next, we take the set of terms $A \subseteq B$ where each $a \in A$ has the largest number of shortest paths between a and q (Equation 3).

$$A = \left\{ x \mid \operatorname*{argmax}_{x \in B} numberOfShortestPaths(x, q) \right\} \tag{3}$$

The correct synonym s has been selected if $s \in A$ and $|A| = 1$. Often the sets A and B will both contain one item, but if $s \in A$ and $|A| > 1$, there is a tie. If $s \notin A$, the selected synonyms are incorrect. If an n-word phrase $c \in C$ is found, its words $c_1, c_2 ..., c_n$ are considered in turn; the c_i closest to q is chosen to represent c. A sought word can be of any part of speech, though only some *WordNet*-based methods allow for adjectives or adverbs, and none can measure distance between different parts of speech. In these problems, we do not consider a word and its morphological variant to be the same.

We generated synonym selection problems specifically for words newly added to *Roget's Thesaurus*. We took all words which appeared either in 1987X5 or in 1911X5, but were not present in the original 1987 or 1911 versions, and used them as query words q for the new problems. We then found in *WordNet* synsets which contain at least one of q's synonyms found in the original (not updated) version of the *Thesaurus*. We completed the problem by finding in the original *Thesaurus* three detractors from q's co-hyponym synsets. This was done for nouns and for verbs, but not for adjectives, for which *WordNet* does not have a strong hypernym hierarchy.

Four different versions of this problem were generated for the 1911 and 1987 *Roget's Thesauri* using nouns and verbs. The linking structure for adjectives in *WordNet* precludes the creation of a data set in this manner. We present the final scores as precision and recall. The precision excludes questions where q is not in *Roget's Thesaurus*, and recall is the score over the entire data set. Precision is thus the proportion of correct guesses out of the questions attempted, while recall is the proportion of correct guesses out of the maximum number of attempted questions. This method of evaluating such work was proposed by Turney (2006).

Table 17 shows the results for nouns and verbs added to both the 1987 and the 1911 versions of *Roget's Thesaurus*. The results are quite similar for all four data sets. Obviously, a precision and recall of 0 is attained for the original versions of the *Thesaurus*. The randomly updated versions did poorly as well. Versions updated after one pass had recall between 18% and 26%, while the versions updated in 5 passes had 40% or more. The random baseline is 25% if all of the questions can be answered. The thesauri updated in 5 passes significantly beat this baseline.[16] The thesauri updated in one pass tended not to show statistically significant improvement, though many problems were unsolvable (q was absent from 1911X1 or 1987X1).

The recall improvement for *Roget's Thesaurus* updated in 5 passes was significantly better (at $p < 0.05$) than for the *Thesaurus* updated in one pass. In turn, the *Thesaurus* updated in one pass was significantly better than the original *Thesaurus* (again at $p < 0.05$). The exception was the 1911 verb data set, for which the improvement could only

[16] Significance was established with Student's T-test with $p < 0.05$.

	Resource	Correct	Wrong	Ties	N/A	Precision	Recall
	1911	0	98	0	98	0	0
1911	1911X1	18	70	10	44	40.13	22.11
Nouns	1911X5	30	45	23	0	39.63	39.63
	1911R	3	93	2	88	39.98	4.08
	1911	0	27	0	27	0	0
1911	1911X1	6	20	1	13	46.42	24.07
Verbs	1911X5	11	14	2	0	44.44	44.44
	1911R	0	27	0	26	0	0
	1987	0	57	0	57	0	0
1987	1987X1	11	38	8	18	38.03	26.02
Nouns	1987X5	18	29	10	0	39.77	39.77
	1987R	0	56	1	52	10.03	0.88
	1987	0	36	0	36	0	0
1987	1987X1	5	27	4	20	41.67	18.52
Verbs	1987X5	12	15	9	0	44.91	44.91
	1987R	1	35	0	29	14.29	2.78

Table 17: Evaluation of identifying synonyms from *WordNet*

be measured as significant at $p < 0.065$. This is largely because the dataset was fairly small. Another observation is that the randomly updated *Thesaurus* only once had a significant improvement over the original *Thesaurus*, in the case of the 1911 noun data set.

These results suggest that the words newly added to *Roget's Thesaurus* are close to the correct location. The newly added words and their synonyms were closer than the newly added words and their cohyponyms. Generally the precision measure showed words added to the 1911X1 and 1987X1 thesauri to be approximately as accurate as, if not slightly more accurate than, those added in passes 2–5. The randomly updated *Thesaurus* did not perform as well, usually falling below the 25% baseline on the precision measure. The results for nouns added to the 1911 *Thesaurus* are a noticeable exception. In the other datasets at most one question was answered correctly by the randomly updated *Thesaurus*, but in this case there were three correct answers. It should be noted, however, that the evaluated sample was very small, so this is likely to have been a coincidence.

6.2 Pseudo-word-sense disambiguation

Pseudo-word-sense disambiguation (PWSD) is a somewhat contrived task, meant to evaluate the quality of a word-sense disambiguation (WSD) system. The set-up for this task is to take two words and merge them into a *pseudo-word*. A WSD system, then, has the goal of identifying which of the two words actually belongs in a given context in which the whole pseudo-word appears. We have had a chance to create a very large dataset for PWSD. This is an opportunity to consider *WordNet* and the versions of *Roget's Thesaurus* in PWSD, and to compare them not only for accuracy but also for runtime.

We used PWSD instead of real WSD for two main reasons. Firstly, as far as we know, there is no WSD data set annotated with *Roget's* word senses and so one would have to be built from scratch. Worse still, to compare WSD systems built using *Roget's Thesaurus* and *WordNet* we would need a dataset labeled with senses from both. Secondly, PWSD gives us a fast way of building a dataset which can be used to evaluate the WSD systems based on the *Thesaurus* and on *WordNet*.

A common variation on this task is to make triples out of a noun and two verbs, then determine which of the verbs takes the noun as its object. The aim is to create a kind of verb disambiguation system which incorporates the edge count distance between nouns. In theory, this measure can help indicate how well a system identifies contexts (verb object) in which a verb appears. That can be useful in real WSD. Others who have worked on variations of PWSD include Gale *et al.* (1992); Schütze (1998); Lee (1999); Dagan *et al.* (1999); Rooth *et al.* (1999); Clark and Weir (2002); Weeds and Weir (2005); Zhitomirsky-Geffet and Dagan (2009). The methodology we followed was similar to that of Weeds and Weir.

The data set was constructed in four steps.

1. Parse Wikipedia with *MINIPAR* (Lin 1998a).
2. Select all object relations and count the frequency of each noun–verb pair $\langle n, v \rangle$.
3. Separate the noun–verb pairs into a training set (80%) and a test set (20%).
4. For each pair $\langle n, v \rangle$ in the test set find another verb v' with the same frequency as v, such that $\langle n, v' \rangle$ appears neither in the training set nor the test set; replace $\langle n, v \rangle$ with the test triple $\langle n, v, v' \rangle$.

This creates two data sets. One is a training set of noun–verb pairs $\langle n, v \rangle$. The other is a test set made up of noun–verb–verb triples $\langle n, v, v' \rangle$. Examples of such triples are ⟨*task, assign, rock*⟩ and ⟨*data, extract, anticipate*⟩. We selected v' so that its frequency is v's frequency $\pm$ 1. We also ensured that the pair $\langle n, v' \rangle$ does not appear anywhere in the training or test data. To reduce noise and decrease the overall size of the dataset, we removed from both the test and training set all noun–verb object pairs which appeared less than five times. This produced a test set of 3327 triples and a training set of 464,303 pairs. We only used half of Wikipedia to generate this data set, the half not used in constructing the noun matrix.

We employed *edgeScore* (Equation 1) for all versions of *Roget's Thesaurus*. The methods implemented in the *WordNet::Similarity* software package (Pedersen *et al.* 2004) determine how close two words are in *WordNet*. These methods are J&C (Jiang and Conrath 1997), Res (Resnik 1995), Lin (Lin 1998a), W&P (Wu and Palmer 1994), L&C (Leacock and Chodorow 1998), H&SO (Hirst and St-Onge 1998), Path (counts edges between synsets), Lesk (Banerjee and Pedersen 2002), and finally Vector and Vector Pair (Patwardhan *et al.* 2003). The measure most similar to the *edgeScore* method is the Path measure in *WordNet*. J&C, Res, Lin, W&P, L&C and Path can only measure relatedness between nouns and verbs, because they only make use of hypernym links. H&SO uses all available *WordNet* relations in finding a path between two words. The Lesk and Vector methods use glosses and so might be just as easily implemented using a dictionary. They need not take advantage of *WordNet's* hierarchical structure.

To perform the PWSD task for each triple $\langle n, v, v' \rangle$, we found in the training corpus k nouns which were the closest to n. Every such noun m got a vote: the number of occurrences of the pair $\langle m, v \rangle$ minus the number of occurrences of $\langle m, v' \rangle$. Any value of k could potentially be used. This means comparing each noun n in the test data to every noun m in the training set if these nouns share a common verb v or v'. Such a computation is feasible in *Roget's Thesaurus*, but it takes a very long time for any *WordNet*-based measure.[17] To ensure that a fair value is

[17] We ran these experiments on an IBM ThinkCenter with a 3.4 GHz Intel Pentium 4 processor and 1.5GB 400 MHz DDR RAM.

Table 18: Pseudo-word-sense disambiguation error rates and run-time

Method	Error Rate	p-value	Change	Time in seconds
1911	0.257	–	–	58
1911X1	0.252	0.000	+1.9%	59
1911X5	0.246	0.000	+4.3%	60
1911R	0.258	0.202	−0.6%	58
1987	0.252	–	–	135
1987X1	0.250	0.152	+0.8%	135
1987X5	0.246	0.010	+2.3%	134
1987R	0.252	0.997	0.0%	134
J&C	0.253	–	–	23 208
Resnik	0.258	–	–	23 112
Lin	0.251	–	–	19 840
W&P	0.245	–	–	38 721
L&C	0.241	–	–	23 445
H&SO	0.257	–	–	2 452 188
Path	0.241	–	–	22 720
Lesk	0.255	–	–	47 625
Vector	0.263	–	–	32 753
Vct Pair	0.272	–	–	74 803

selected, we divided the test set into 30 sets. We use 29 folds to find the optimal value of k and apply it to the 30^{th} fold.

The score for the PWSD task is typically measured as an error rate where T is the number of test cases (Equation 4).

$$\textit{Error rate} = \frac{1}{T}\left(\#\ \textit{incorrect choices} + \frac{\#\ \textit{ties}}{2}\right) \tag{4}$$

Table 18 shows the results of this experiment. The improvement on 1911X1 and 1911X5 over the original 1911 version of the *Thesaurus* was statistically significant at $p < 0.05$, according to Student's T-test. The improvement on the updated 1987 version was not statistically significant for 1987X1 with $p \approx 0.15$, but it was significant for 1987X5. The 1911X5 version gave results comparable to the 1987 version. The *Roget's*-based methods were comparable to the best *WordNet*-based methods.

When it comes to the values of k, $k = 1$ was always found to be the optimal value on this dataset. So, the best way to perform PWSD

is to select the *nearest* noun taken as the object of either v or v'.

The CPU usage was perhaps the most pronounced difference with *Roget's*-based methods, which ran in a tiny fraction of the time which *WordNet*-based methods required. H&SO took around 28 days to run, so this measure simply is not an option for large-scale semantic relatedness problems. Even Lin, the fastest *WordNet*-based method, took around 5.5 hours, over 340 times longer than than the method based on the 1911 *Thesaurus*.

For all systems, a total of 193192 word pairs must be compared. We also examined the number of necessary comparisons between word senses. If one resource contains a larger number of senses of each word it is measuring distance on, then it will necessarily have to perform many more comparisons. The method based on the 1987 *Thesaurus* required nearly 120 million comparisons. The method based on the 1911 *Thesaurus* needed 14.7 million comparisons. For the *WordNet*-based methods only 3.5 million comparisons were necessary. Clearly the implementation of *Roget's Thesaurus* has a very strong advantage when it comes to runtime.

6.3 *SAT analogies*

The last class of problems to which we applied *Roget's Thesaurus* were analogy problems in the style of Scholastic Aptitude Tests (SAT). In an SAT analogy task, one is given a *target pair* $\langle A, B \rangle$ and then from a list of possible candidates selects the pair $\langle C, D \rangle$ most similar to the target pair. Ideally, the relation between the pair $\langle A, B \rangle$ should be the same as the relation between the pair $\langle C, D \rangle$. For example:

Target pair	*word, language*
Candidates	*paint, portrait*
	poetry, rhythm
	note, music
	tale, story
	week, year

Roget's Thesaurus performs well on problems of selecting synonyms and pseudo-word-sense disambiguation, but it is not clear just

Table 19: Scores in the analogy problem solved by matching kinds of relations (P = precision, R = recall, WN = *WordNet*)

System	Correct	Ties	Incorrect	Filtered	P	R	F1
1911	14	21	39	300	0.307	0.061	0.102
1911X1	15	23	39	297	0.321	0.066	0.110
1911X5	15	27	39	293	0.330	0.072	0.118
1911R	14	21	39	300	0.307	0.061	0.102
1987	18	85	81	190	0.271	0.133	0.179
1987X1	19	85	81	189	0.273	0.135	0.181
1987X5	21	85	81	187	0.278	0.139	0.185
1987R	18	86	80	190	0.271	0.133	0.179
WN 3.0	20	4	12	338	0.600	0.058	0.105

how well it will do on tasks of identifying analogies. That is because relations in the *Thesaurus* are unlabelled. We explore two methods of solving such problems with both the *Thesaurus* and *WordNet*. The first method attempts to identify a few kinds of relations in the *Thesaurus* and then apply them to identifying analogies. The second method uses edge distance between the pairs $\langle A,B\rangle$–$\langle C,D\rangle$ and $\langle A,C\rangle$–$\langle B,D\rangle$ as a heuristic for guessing whether two word pairs contain the same relation.

The dataset contains 374 analogy problems extracted from real SAT tests and practice tests (Turney 2005). A problem contains a *target pair* $\langle A,B\rangle$ and several pairs to choose from: $test_i = \langle X_i, Y_i\rangle$, $i = 1..5$. In evaluation, we consider seven scores: correct, ties, incorrect, filtered out, precision, recall and equal-weighted F-score. We define precision and recall in the same way as in Section 6.1. In the case of an n-way tie, the correct answer counts as $1/n$ towards the precision and recall. We consider recall as the most important measure, because it evaluates each method over the entire data set.

6.3.1 Matching relations

Unlike *WordNet*, *Roget's Thesaurus* contains no explicitly labelled semantic relations, but certain implicit relations can be inferred from its structure. Near-synonyms tend to appear in the same SG. Near-antonyms usually appear in different Heads in the same Head Group. One can also infer a hierarchical relation between two words if (1) they are in the same Paragraph and one of them is in the first SG, or (2) they are in the same POS and one of them is in the first SG of the

first Paragraph. So, three relations can be deduced from the *Thesaurus*. Two words can be near-synonymous, near-antonymous or hierarchically related. From *WordNet*, we allow words to be related by any of the explicit semantic relations. We also apply hypernymy/hyponymy transitively.

Using these semantic relations, the analogy problem is solved by identifying a candidate analogy which contains the same relation as the target pair. There will be no solution if no relation can be found for the target pair. This experiment is interesting in that it helps test whether narrower semantic relations in *WordNet* are more useful or less useful than the broader relations in *Roget's Thesaurus*. Table 19 shows the results; "Filtered" shows the number of pairs which were not scored because no relation could be established between the words in the target or candidate pairs.

The *WordNet*-based method has high precision, but recall is low compared to that of the *Roget's*-based versions. Interestingly, the precision and recall both increase as more words are added to the 1911 and 1987 versions of *Roget's*. We consider recall as more important in this evaluation, so it is clear that the most updated versions of *Roget's Thesaurus* outperform *WordNet* by a fair margin. Although the original 1911 version gave a lower F-score than *WordNet*, all other versions performed better. The existence of very specific semantic relations in *WordNet* did give it an edge in precision, but *WordNet* was only able to answer a few questions. This suggests that the relations between pairs in analogy tests are not only of the type encountered in *WordNet*. While the broader relations identified in the *Thesaurus* appear to be less reliable and give lower precision, the recall is much higher.

6.3.2 Edge distance

The second method of solving analogy problems uses edge distance between words as a heuristic. Analogy problems have been solved in this way using Equation 5 proposed by Turney (2006).

$$score(\langle A, B\rangle : \langle X_i, Y_i\rangle) = \frac{1}{2}\big(sim_a(A, X_i) + sim_a(B, Y_i)\big) \qquad (5)$$

The highest-scoring pair $\langle X_i, Y_i\rangle$ is guessed as the correct analogy. This method assumes that A and X_i should be closely related and

so should B and Y_i. An illustrative example is $\langle carpenter,\ wood \rangle$ and $\langle mason,\ stone \rangle$.

In Equation 5, sim_a is the attributional similarity. We replaced it with an edge distance measure r, either *edgeScore* (Equation 1) or one of the measures built on *WordNet*. Because *edgeScore* only returns even numbers between 0 and 18, it tends to give many ties. We used a formula with a tie breaker based on the edge distance between A and B and between X_i and Y_i:[18]

$$score(\langle A,B \rangle, \langle X_i, Y_i \rangle) = r(A,X_i) + r(B,Y_i) + \frac{1}{|r(A,B) - r(X_i,Y_i)| + 1} \qquad (6)$$

The last term of the sum in Equation 6 acts as a tie-breaker which favours candidates $\langle X_i,\ Y_i \rangle$ with an edge distance similar to the target $\langle A,\ B \rangle$. We include another constraint: A and X_i must be in the same part of speech, and so do B and Y_i. Only one sense of each of A, B, X_i and Y_i can be used in the calculation of Equation 6. For example, the same sense of A is used when calculating $r(A,\ X_i)$ and $r(A,\ B)$.

We applied Equation 6 to the 374 analogy problems using all versions of *Roget's Thesaurus* and the *WordNet*-based edge distance measures. The results are shown in Table 20. The "Filtered" column shows how many SAT problems could not be solved because at least one of the words needed was absent in either the *Thesaurus* or *WordNet*. Unfortunately, expanding *Roget's Thesaurus* did not reduce the number of filtered results. That said, both precision and recall increased when more words were added to the *Thesaurus*. Overall, we found that in absolute numbers the updated 1987X5 *Roget's Thesaurus* performed better than any other resource examined. Even the updated versions of the 1911 *Thesaurus* performed on par with the best *WordNet*-based systems. We must note, however, that none of the improvements of the 1987X5-based method over any given *WordNet* method are statistically significant.

[18] We owe this formula to a personal communication with Dr. Vivi Nastase. It was also used in (Kennedy and Szpakowicz 2007)

System	Correct	Ties	Misses	Filtered	P	R	F1
1911	98	11	214	51	0.319	0.276	0.296
1911X1	98	17	208	51	0.329	0.284	0.305
1911X5	97	20	206	51	0.330	0.285	0.306
1911R	97	12	218	47	0.313	0.274	0.292
1987	101	35	232	6	0.318	0.313	0.316
1987X1	102	38	228	6	0.324	0.319	0.322
1987X5	102	39	227	6	0.325	0.320	0.323
1987R	103	34	233	4	0.320	0.316	0.318
Path	85	5	166	118	0.342	0.234	0.278
J&C	80	0	176	118	0.312	0.214	0.254
Resnik	91	16	149	118	0.385	0.263	0.313
Lin	82	3	171	118	0.325	0.222	0.264
W&P	90	1	165	118	0.354	0.242	0.287
L&C	91	4	161	118	0.363	0.249	0.295
H&SO	96	39	212	27	0.321	0.298	0.309
Lesk	113	0	234	27	0.326	0.302	0.313
Vector	113	0	234	27	0.326	0.302	0.313
Vector Pair	106	0	241	27	0.305	0.283	0.294

Table 20: Scores in the analogy problem solved using semantic distance functions (P = precision, R = recall)

7 CONCLUSION

7.1 *Summary*

We have described a method of automatically updating *Roget's Thesaurus* with new words. The process has two main steps: lists of semantically related words are generated, and next those lists are used to find a place for a new word in the *Thesaurus*. We have enhanced both steps by leveraging the structure of *Roget's Thesaurus*.

When creating lists of related words, we have evaluated a technique for measuring semantic relatedness which enhances distributional methods using lists of known synonyms. We have shown this to have a statistically significant effect on the quality of measures of semantic relatedness.

In the second step, the actual addition of new words to *Roget's Thesaurus*, we generated a list of neighbouring words and used them as anchors to identify where in the *Thesaurus* to place a new word. This process benefits from tuning on the actual *Thesaurus*. The task here is to find words which will be good anchors for determining where to

place a new term. We experimented with three methods of finding anchors, using the rank, the relatedness score and a relative relatedness score. We found that rank worked best. The process of adding new words to *Roget's Thesaurus* is hierarchical. First the Part of Speech in the *Thesaurus* is identified, then the Paragraph, then the Semicolon Group. A new Paragraph or Semicolon Group can be created if needed.

A manual evaluation of our methodology found that added words were almost indistinguishable from words already present in the *Thesaurus*. Even after multiple passes the words seemed to find fairly accurate placing in an existing Paragraph. When adding words to a new Paragraph, after one pass the words were highly accurate, but the accuracy fell after additional passes. In total, some 5500 words were added to the 1911 version and some 9600 words to the 1987 version.

We also performed an application-based evaluation to compare the original and updated versions of *Roget's Thesaurus* and, when possible, *WordNet*. The tasks were synonym identification, pseudo-word-sense disambiguation and SAT-style analogy problems. On all tasks the updates to the *Thesaurus* showed a noticeable improvement. In our evaluations, *Roget's Thesaurus* also performed as well as, or better than, *WordNet*. In particular, it could perform calculations many times faster than the *WordNet::Similarity* software package (Pedersen *et al.* 2004).

Most of our experiments show that the 1987 version of *Roget's Thesaurus* outperforms the 1911 version. There are two reasons. First, for our measure of semantic relatedness, the evaluation was conducted on words in the same *Roget's grouping* from the 1987 version. Since the structure of the *Thesaurus* is used to train our MSR, it is natural that scores are higher when training and evaluation are done on the same version. The second reason is simply that the 1987 version is larger. When adding new words to *Roget's Thesaurus*, a larger thesaurus gives more potential anchor words to help find an appropriate placement for a new word. For our task-based evaluation, the applications we chose will naturally benefit from a larger thesaurus as well.

7.2 *Future work*

The supervised measure of semantic relatedness provides an interesting method of re-weighting contexts. Recent work has shown that similar techniques make it possible to find a weighted mapping be-

tween the context space in two different languages (Kennedy and Hirst 2012). Methods of this kind could be used to emphasise similarities between words based on sentiment, emotion or formality, rather than simply on synonymy. Using emotionally related words as a source of training data could enable the creation of a measure of semantic relatedness which favours words of the same emotional class over other, nearer, synonyms conveying a different emotion.

Perhaps other more complex methods of adding new words to *Roget's Thesaurus* can be considered. For example, mixing rank and score (maybe using machine learning) might lead to an even more accurate method. Other methods of identifying where in the *Thesaurus* to place a word could also be considered. In particular, Pantel's (2005) method could potentially be modified to work for *Roget's Thesaurus*.

Our method only adds individual words to *Roget's Thesaurus*. It should be possible to expand it into adding multi-word phrases. Many dependency parsers can identify noun phrases and so can be used to create distributional vectors for such phrases. Adding multi-word phrases to verb or adjective *Roget's groupings* may be possible by identifying n-grams which are frequent in a text. Two problems arise. One is determining whether high frequency alone is a good enough reason to add a multi-word phrase. The second is how to represent such multi-word phrases. It could be possible to represent them by vectors of word–relation pairs for syntactically related words in the same sentence, but outside of the phrase being considered. The meaning of a phrase may also be deduced by composing the distributional vectors of its individual words. There is ongoing, and very interesting, research in this area (Mitchell and Lapata 2008; Turney 2012).

A problem which we have not tackled yet is that of adding cross-references: if the same word appears in two places in *Roget's Thesaurus*, then often a cross-reference links the two occurrences. Making use of these cross-references could be a considerable undertaking, because it requires, amongst other things, some form of effective word-sense disambiguation.

The manual annotation has only been conducted on the 1911 version of *Roget's Thesaurus*, because it is the only version which can be released to the public, and because the annotation experiment has been very time-consuming. In the interest of completeness, the updates to the 1987 version could be evaluated similarly. We expect that those

updates should actually be more accurate, because the 1911 version is both older and smaller. This would be in line with the automatic evaluation from Section 4, but it is yet to be confirmed manually.

It should be possible to adapt our methods of placing words in *Roget's Thesaurus* to work for *WordNet*. Instead of identifying words in the same POS, then Paragraph, then SG, word groupings could be created from *WordNet*'s hypernym hierarchy. We envisage two ways of doing this. The first would be to pick a relatively high level within the hierarchy and classify each word into one or more of the synsets at that level, much as we did with the POS level. A synset could be represented by all the words in the transitive closure of its hyponym synsets. Next, the word would be propagated down the hierarchy – as we do with Paragraphs and SGs – until it can go no further, and then added to the synset there.

This method could not (yet) be applied to adjectives, and would only take one kind of relation into account when placing a word in *WordNet*. Another option is to create a neighbourhood of words for each synset, based on a variety of relations. A word could then be placed in a larger grouping of multiple synsets before the particular synset it belongs to is determined. If no synset can be picked, then a new synset can be created with some sort of ambiguous link joining it to the other synsets in its neighbourhood. A hybrid of these two methods is also possible. Our first method could be enhanced by using not only a synset's terms, but also its close neighbours. This would expand the set of anchor words at the cost of introducing words common to multiple synsets.

It should also be possible to port our method to thesauri and wordnets in other languages. The main problem might be our method's reliance on a dependency parser. Such parsers are not available yet for many languages. Nonetheless, it could be possible to replicate much of the relevant functionality of a dependency parser using a part-of-speech tagger – and taggers are quite widely available. For example, one may assume that a noun can only be modified by other nouns or adjectives in its vicinity, and so only use those terms in constructing a distributional vector.

Another direction which this kind of research could take would be to test the methods on adding words in a particular domain. Most of the words in *Roget's Thesaurus* are from everyday English, as op-

posed to, say, medical terms. The nearest synonyms of such technical words will be other technical words. This could make it more difficult to actually add domain-specific terms to *Roget's Thesaurus*. That said, the trainable measure of semantic relatedness from Kennedy and Szpakowicz (2011, 2012) could be built using words of a particular domain. If domain-specific and everyday words could be grouped as near-synonyms, then an MSR could be trained for adding domain-specific terms to *Roget's Thesaurus*.

Similar to adding domain-specific words is the challenge of adding brand new coinage to *Roget's Thesaurus*. Very new words may not have close synonyms in the *Thesaurus*, which is why we add words in multiple passes. It would be interesting to investigate how many passes are required before, say, the word "iPhone" is added to the *Thesaurus*. Closely related phrases like "mobile phone" or "smart phone" would need to already be present. Other terms, such as "cellular network", "texting" or "Apple", could also be useful in choosing where to place a word like "iPhone".

Finally, note that we have only applied *Roget's Thesaurus* to three NLP tasks, to demonstrate value in both its structure and language coverage. Many other applications of the *Thesaurus* are possible. Some obvious ones include real word-sense disambiguation and lexical substitution. *Roget's Thesaurus* has already been used in the construction of lexical chains (Morris and Hirst 1991; Jarmasz and Szpakowicz 2003a). Lexical chains might be applied to summarisation or text segmentation. Since the *Thesaurus* contains a large number of opposing concepts, it may be possible to apply it to lexical entailment as well.

NLP researchers are always on the hunt for newer and larger data sets on which to train and evaluate their experiments. Many of these experiments will require measuring semantic distance among huge sets of words. In the coming years, the trend towards analyzing big data will drive the need for fast semantic relatedness calculation. *Roget's Thesaurus* is uniquely suited for that.

REFERENCES

Satanjeev BANERJEE and Ted PEDERSEN (2002), An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet, in *Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics – CICLing 2002*, pp. 136–145, Mexico City, Mexico.

Patrick J. CASSIDY (2000), An Investigation of the Semantic Relations in the Roget's Thesaurus: Preliminary Results, in *Proceedings of the Conference on Intelligent Text Processing and Computational Linguistics – CICLing 2000*, pp. 181–204, Mexico City, Mexico.

Stephen CLARK and David WEIR (2002), Class-based Probability Estimation Using a Semantic Hierarchy, *Computational Linguistics*, 28(2):187–206.

Carolyn J. CROUCH (1988), A Cluster-based Approach to Thesaurus Construction, in *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval – SIGIR 1988*, pp. 309–320, Grenoble, France.

Carolyn J. CROUCH and Bokyung YANG (1992), Experiments in Automatic Statistical Thesaurus Construction, in *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval – SIGIR 1992*, pp. 77–88, Copenhagen, Denmark.

DAGAN, IDO, Lillian LEE, and Fernando PEREIRA (1999), Similarity-based Models of Word Co-occurrence Probabilities, *Machine Learning*, 34(1-3):43–69.

Andrea ESULI and Fabrizio SEBASTIANI (2006), SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining, in *Proceedings of the 5th Conference on Language Resources and Evaluation – LREC 2006*, pp. 417–422, Genoa, Italy.

Christiane FELLBAUM, editor (1998), *WordNet: an Electronic Lexical Database*, MIT Press, Cambridge, MA, USA.

William A. GALE, Kenneth W. CHURCH, and David YAROWSKY (1992), A Method for Disambiguating Word Senses in a Large Corpus, *Computers and the Humanities*, 26:415–439.

Roxana GIRJU, Adriana BADULESCU, and Dan MOLDOVAN (2003), Learning Semantic Constraints for the Automatic Discovery of Part–Whole Relations, in *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics – HLT-NAACL 2003*, pp. 1–8, Edmonton, Canada.

Roxana GIRJU, Adriana BADULESCU, and Dan MOLDOVAN (2006), Automatic Discovery of Part–Whole Relations, *Computational Linguistics*, 32(1):83–136.

Marti A. HEARST (1992), Automatic Acquisition of Hyponyms from Large Text Corpora, in *Proceedings of the 14th International Conference on Computational Linguistics – COLING 1992*, pp. 539–545, Nantes, France.

Graeme HIRST and David ST-ONGE (1998), Lexical Chains as Representation of Context for the Detection and Correction of Malapropisms, in Christiane FELLBAUM, editor, *WordNet: An Electronic Lexical Database*, pp. 305–322, MIT Press, Cambridge, MA, USA.

Mario JARMASZ (2003), *Roget's Thesaurus as a Lexical Resource for Natural Language Processing*, Master's thesis, University of Ottawa, Canada.

Mario JARMASZ and Stan SZPAKOWICZ (2003a), Not as Easy As It Seems: Automating the Construction of Lexical Chains Using Roget's Thesaurus, in *16th Conference of the Canadian Society for Computational Studies of Intelligence – AI 2003, Halifax, Canada*, number 2671 in Lecture Notes in Computer Science, pp. 544–549, Springer, Berlin/Heidelberg, Germany.

Mario JARMASZ and Stan SZPAKOWICZ (2003b), Roget's Thesaurus and Semantic Similarity, in *Proceedings of the Conference on Recent Advances in Natural Language Processing – RANLP 2003*, pp. 212–219, Borovets, Bulgaria.

Mario JARMASZ and Stan SZPAKOWICZ (2004), Roget's Thesaurus and Semantic Similarity, in N. NICOLOV, K. BONTCHEVA, G. ANGELOVA, and R. MITKOV, editors, *Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003*, volume 260 of *Current Issues in Linguistic Theory*, pp. 111–120, John Benjamins, Amsterdam, The Netherlands/Philadelphia, PA, USA.

Jay J. JIANG and David W. CONRATH (1997), Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy, in *Proceedings of the 10th International Conference on Research on Computational Linguistics – ROCLING X*, pp. 19–33, Taipei, Taiwan.

Joshua C. KENDALL (2008), *The Man Who Made Lists : Love, Death, Madness, and the Creation of Roget's Thesaurus*, G.P.Putnam's Sons, New York, NY, USA.

Alistair KENNEDY and Graeme HIRST (2012), Measuring Semantic Relatedness Across Languages, in *xLiTe: Cross-Lingual Technologies, workshop collocated with the Conference on Neural Information Processing Systems – NIPS 2012*, Lake Tahoe, NV, USA.

Alistair KENNEDY and Stan SZPAKOWICZ (2007), Disambiguating Hypernym Relations for Roget's Thesaurus, in *Proceedings of the 10th International Conference on Text, Speech and Dialogue – TSD 2007, Pilsen, Czech Republic*, number 4629 in Lecture Notes in Artificial Intelligence, pp. 66–75, Springer, Berlin/Heidelberg, Germany.

Alistair KENNEDY and Stan SZPAKOWICZ (2011), A Supervised Method of Feature Weighting for Measuring Semantic Relatedness, in *Proceedings of the Canadian Conference on Artificial Intelligence – AI 2011, St. John's, Canada*, number 6657 in Lecture Notes in Artificial Intelligence, pp. 222–233, Springer, Berlin/Heidelberg, Germany.

Alistair KENNEDY and Stan SZPAKOWICZ (2012), Supervised Distributional Semantic Relatedness, in *Proceedings of the 15th International Conference on Text, Speech and Dialogue – TSD 2012, Brno, Czech Republic*, number 7499 in Lecture Notes in Artificial Intelligence, pp. 207–214, Springer, Berlin/Heidelberg, Germany.

Betty KIRKPATRICK, editor (1987), *Roget's Thesaurus of English Words and Phrases*, Longman, Harlow, UK.

Klaus KRIPPENDORFF (2004), *Content Analysis: An Introduction to Its Methodology*, Sage Publications Inc., Los Angeles, CA, USA, 2nd edition.

Oi Yee KWONG (1998a), Aligning WordNet with Additional Lexical Resources, in *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, pp. 73–79, Montréal, Canada.

Oi Yee KWONG (1998b), Bridging the Gap Between Dictionary and Thesaurus, in *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics – ACL 1998*, pp. 1487–1489, Montréal, Canada.

Robin J. LANDIS and G. G. KOCH (1977), The Measurement of Observer Agreement for Categorical Data, *Biometrics*, 33:159–174.

Claudia LEACOCK and Martin CHODOROW (1998), Combining Local Context and WordNet Sense Similarity for Word Sense Disambiguation, in Christiane FELLBAUM, editor, *WordNet: An Electronic Lexical Database*, pp. 265–284, MIT Press, Cambridge, MA, USA.

Lillian LEE (1999), Measures of Distributional Similarity, in *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics – ACL 1999*, pp. 25–32, College Park, MD, USA.

Lothar LEMNITZER, Holger WUNSCH, and Piklu GUPTA (2008), Enriching GermaNet with Verb–Noun Relations – a Case Study of Lexical Acquisition, in *Proceedings of the 6th International Conference on Language Resources and Evaluation – LREC 2008*, Marrakech, Morocco.

Dekang LIN (1998a), Automatic Retrieval and Clustering of Similar Words, in *Proceedings of the 17th International Conference on Computational Linguistics – COLING 1998*, pp. 768–774, Montréal, Canada.

Dekang LIN (1998b), Dependency-Based Evaluation of MINIPAR, in *Proceedings of the Workshop on the Evaluation of Parsing Systems at the 1st International Conference on Language Resources and Evaluation – LREC 1998*, Granada, Spain.

Bernardo MAGNINI and Gabriela CAVAGLIÁ (2000), Integrating Subject Field Codes into WordNet, in *Proceedings of the 2nd International Conference on Language Resources and Evaluation – LREC 2000*, pp. 1413–1418, Athens, Greece.

Jeff MITCHELL and Mirella LAPATA (2008), Vector-based models of semantic composition, in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies – ACL 2008: HLT*, pp. 236–244, Columbus, OH, USA.

Emmanuel MORIN and Christian JACQUEMIN (1999), Projecting Corpus-Based Semantic Links on a Thesaurus, in *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics – ACL 1999*, pp. 389–396, College Park, MD, USA.

Jane MORRIS and Graeme HIRST (1991), Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text, *Computational Linguistics*, 17(1):21–48.

Vivi NASTASE and Stan SZPAKOWICZ (2001), Word Sense Disambiguation in Roget's Thesaurus Using WordNet, in *Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources*, pp. 12–22, Pittsburgh, PA, USA.

Patrick PANTEL (2003), *Clustering by Committee*, Ph.D. thesis, University of Alberta, Canada.

Patrick PANTEL (2005), Inducing Ontological Co-occurrence Vectors, in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics – ACL 2005*, pp. 125–132, Ann Arbor, MI, USA.

Siddharth PATWARDHAN, Satanjeev BANERJEE, and Ted PEDERSEN (2003), Using Measures of Semantic Relatedness for Word Sense Disambiguation, in *Proceedings of the 4th International Conference on Intelligent Text Processing and Computational Linguistics – CICLing-2003*, pp. 241–257, Mexico City, Mexico.

Ted PEDERSEN, Siddhart PATWARDHAN, and Jason MICHELIZZI (2004), Wordnet::Similarity - Measuring the Relatedness of Concepts, in *Proceedings of the 19th National Conference on Artificial Intelligence – AAAI 2004*, pp. 1024–1025, San Jose, CA, USA.

Maciej PIASECKI, Bartosz BRODA, Michał MARCIŃCZUK, and Stan SZPAKOWICZ (2009), The WordNet Weaver: Multi-criteria Voting for Semi-automatic Extension of a Wordnet, in *Proceedings of the 22nd Canadian Conference on Artificial Intelligence – AI 2009, Kelowna, Canada*, number 5549 in Lecture Notes in Artificial Intelligence, pp. 237–240, Springer, Berlin/Heidelberg, Germany.

Paul PROCTER (1978), *Longman Dictionary of Contemporary English*, Longman Group Ltd., Essex, UK.

Philip RESNIK (1995), Using Information Content to Evaluate Semantic Similarity, in *Proceedings of the 14th International Joint Conference on Artificial Intelligence – IJCAI 1995*, pp. 448–453, Montréal, Canada.

Mats ROOTH, Stefan RIEZLER, Detlef PRESCHER, Glenn CARROLL, and Franz BEIL (1999), Inducing a Semantically Annotated Lexicon via EM-based Clustering, in *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics – ACL 1999*, pp. 104–111, College Park, MD, USA.

Sara RYDIN (2002), Building a Hyponymy Lexicon with Hierarchical Structure, in *Proceedings of the ACL-02/SIGLEX Workshop on Unsupervised Lexical Acquisition – ULA 2002*, pp. 26–33, Philadelphia, PA, USA.

Benoît SAGOT and Darja FIŠER (2011), Extending wordnets by learning from multiple resources, in *Proceedings of the 5th Language and Technology Conference – LTC 2011*, pp. 526–530, Poznań, Poland.

Erik Tjong Kim SANG (2007), Extracting Hypernym Pairs from the Web, in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics – ACL 2007 (Interactive Poster and Demonstration Sessions)*, pp. 165–168, Prague, Czech Republic.

Hinrich SCHÜTZE (1998), Automatic Word Sense Discrimination, *Computational Linguistics*, 24(1):97–123.

Keiji SHINZATO and Kentaro TORISAWA (2004), Acquiring Hyponymy Relations from Web Documents, in *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics – HLT-NAACL 2004*, pp. 73–80, Boston, MA, USA.

Rion SNOW, Daniel JURAFSKY, and Andrew Y. NG (2005), Learning Syntactic Patterns for Automatic Hypernym Discovery, in Lawrence K. SAUL, Yair WEISS, and Léon BOTTOU, editors, *Advances in Neural Information Processing Systems 17*, pp. 1297–1304, MIT Press, Cambridge, MA, USA.

Rion SNOW, Daniel JURAFSKY, and Andrew Y. NG (2006), Semantic Taxonomy Induction from Heterogenous Evidence, in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics – COLING/ACL 2006*, Sydney, Australia.

Ratanachai SOMBATSRISOMBOON, Yutaka MATSUO, and Mitsuru ISHIZUKA (2003), Acquisition of Hypernyms and Hyponyms from the WWW, in *Procedings of the 2nd International Workshop on Active Mining – AM 2003 (in Conjunction with the International Symposium on Methodologies for Intelligent Systems)*, pp. 7–13, Maebashi City, Japan.

Carlo STRAPPARAVA and Alessandro VALITUTTI (2004), WordNet-Affect: an Affective Extension of WordNet, in *Proceedings of the 4th International Conference on Language Resources and Evaluation – LREC 2004*, pp. 1083–1086, Lisbon, Portugal.

Hiroaki TSURUMARU, Toru HITAKA, and Sho YOSHIDA (1986), An Attempt to Automatic Thesaurus Construction from an Ordinary Japanese Language Dictionary, in *Proceedings of the 11th Conference on Computational Linguistics – COLING 1986*, pp. 445–447, Bonn, Germany.

Peter TURNEY (2005), Measuring Semantic Similarity by Latent Relational Analysis, in *Proceedings of the 19th International Joint Conference on Artificial Intelligence – IJCAI-05*, pp. 1136–1141, Edinburgh, Scotland.

Peter TURNEY (2006), Similarity of Semantic Relations, *Computational Linguistics*, 32(3):379–416.

Peter TURNEY (2012), Domain and Function: A Dual-Space Model of Semantic Relations and Compositions, *Journal of Artificial Intelligence Research*, 44:533–585.

Piek VOSSEN, editor (1998), *EuroWordNet: a Multilingual Database with Lexical Semantic Networks*, Kluwer Academic Publishers, Norwell, MA, USA.

Julie WEEDS and David WEIR (2005), Co-occurrence Retrieval: A Flexible Framework for Lexical Distributional Similarity, *Computational Linguistics*, 31(4):439–475.

Zhibiao WU and Martha PALMER (1994), Verb Semantics and Lexical Selection, in *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics – ACL 1994*, pp. 133–138, Las Cruces, NM, USA.

Hao ZHENG, Xian WU, and Yong YU (2008), Enriching WordNet with Folksonomies, in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining – PAKDD 2008*, number 5012 in Lecture Notes in Artificial Intelligence, pp. 1075–1080, Springer, Berlin/Heidelberg, Germany.

Maayan ZHITOMIRSKY-GEFFET and Ido DAGAN (2009), Bootstrapping distributional feature vector quality, *Computational Linguistics*, 35(3):435–461.