

# Disambiguating Hypernym Relations for *Roget's* Thesaurus

Alistair Kennedy<sup>1</sup> and Stan Szpakowicz<sup>1,2</sup>

<sup>1</sup> School of Information Technology and Engineering, University of Ottawa  
Ottawa, Ontario, Canada

akennedy@site.uottawa.ca, szpak@site.uottawa.ca

<sup>2</sup> Institute of Computer Science, Polish Academy of Sciences  
Warsaw, Poland

**Abstract.** *Roget's* Thesaurus is a lexical resource which groups terms by semantic relatedness. It is *Roget's* shortcoming that the relations are ambiguous, in that it does not *name* them; it only shows that there *is* a relation between terms. Our work focuses on disambiguating hypernym relations within *Roget's* Thesaurus. Several techniques of identifying hypernym relations are compared and contrasted in this paper, and a total of over 50,000 hypernym relations have been disambiguated within *Roget's*. Human judges have evaluated the quality of our disambiguation techniques, and we have demonstrated on several applications the usefulness of the disambiguated relations.

## 1 Introduction

*Roget's* Thesaurus has proven useful in several applications, including determining semantic similarity between terms [1]. *Roget's* is a good resource for Natural Language Processing, not the least because it contains many terms and phrases not found in other lexical resources. One factor limits the usefulness of *Roget's*: unlike in *WordNet* [2], the relations between terms are not named. Instead, *Roget's* clusters terms according to certain kinds of implicit semantic relatedness. Although it is usually clear to people that words in the Thesaurus are related, it is not always clear in what way. In this paper, we describe methods of disambiguating hypernym relations in *Roget's* Thesaurus. To demonstrate that this is useful, we show how these relations can improve *Roget's* capacity for solving problems of semantic similarity, synonym identification and analogy identification. We work with the 1987 version of *Penguin's Roget's Thesaurus* [3].

### 1.1 Semantic Distances in *Roget's* Thesaurus

*Roget's* Thesaurus has been implemented in Java as an Electronic Lexical Knowledge Base (ELKB) [4]. An 8-level hierarchy for grouping words and phrases in the Thesaurus induces a measure of semantic distance between words/phrases [1]. A distance is calculated as the length of the shortest path through the hierarchy between two given terms. A score reflects the level at which both words/phrases appear. The Semicolon Group contains the most closely related terms, while the Class is the broadest category:

- distance 0 – the same *Semicolon Group*
- distance 2 – the same *Paragraph*
- distance 4 – the same *Part of Speech*
- distance 6 – the same *Head*
- distance 8 – the same *Head Group*
- distance 10 – the same *Sub-Section*
- distance 12 – the same *Section*
- distance 14 – the same *Class*
- distance 16 – different *Classes*, or a word or phrase not found

The Part of Speech group found in *Roget's* Thesaurus does not contain all terms/phrases within a particular part of speech, only those terms of a given POS related to a particular subject (Head). There can also be cross references between Heads in the Thesaurus. An example of a paragraph appears in Figure 1. Each line is a semicolon group.

support, underpinning, (703 aid);  
 leg to stand on, point d'appui, footing, ground, terra firma;  
 hold, foothold, handhold, toe-hold, (778 retention);  
 life jacket, lifebelt, (662 safeguard);  
 life-support machine or system;

**Fig. 1.** The first paragraph from Head 218

## 1.2 Related Work on Discovering Hypernyms

It is a time-consuming task to construct a large lexical resource that would be as trustworthy as *WordNet*: much work must be done manually. In recent years there has been research on ways to construct such lexical resources automatically from a corpus, in particular by creating hypernym hierarchies. Often people apply patterns similar to those proposed by Hearst [5], with modifications to improve precision and recall [6,7,8]. People have also considered Machine Learning in the identification of hypernyms in text [9], and mined dictionaries for relations [10], including relations other than hypernyms. In recent years some systems, such as Espresso [11], have been designed to identify a variety of different semantic relations from text. Similar research has been done on labeling semantic classes using *is a* relations [12].

## 2 Potential Relations in *Roget's* Thesaurus

We need to know where in *Roget's* hierarchy we can generally encounter hypernymy. To find out, we took relations from *WordNet* and counted how many of them mapped to *Roget's* Thesaurus at various levels of granularity. We decided that relations would have to be between terms/phrases in the same Semicolon Group, Paragraph or Part of Speech. This eliminates the need for word sense disambiguation (into *Roget's* word senses), since the same word in two different senses rarely appears in the same Part of Speech. We found a total of 57,478 relations in the same Part of Speech, 45,481 in the same Paragraph and 15,106 in the same Semicolon Group. We found relatively few

relations in the same Semicolon Group, compared to the Paragraph and Part of Speech. Since about 80% of all relations found at the Part of Speech level also appeared in the same Paragraph, we chose to focus on disambiguating relations in the Paragraph.

### 3 Identifying Relations

To identify hypernym relations in *Roget's* Thesaurus, we look at a variety of resources, using a variety of techniques. For each of the identified hypernym relations we attempted to find all the places where this relation appears in *Roget's*. If both terms in the relation are found to be in the same *Roget's* paragraph, the relation in *Roget's* is disambiguated and labeled as a hypernym relation. To accomplish this effectively, the ELKB is used to generate all morphological forms of the terms. This is necessary since many words in *Roget's* are not in their base form.

*Roget's* Thesaurus is a large resource, with over 50,000 unique nouns and noun phrases, many of them absent from other lexical resources or corpora. We applied three different methods of identifying hypernym/hyponym relations: including lexical resources – *WordNet* and *OpenCyc*; search in dictionaries – Longman Dictionary of Contemporary English (LDOCE) and Wiktionary; and examine large corpora using patterns proposed by Hearst [5] – the British National Corpus (BNC) and the Waterloo MultiText (WMT) corpus. It is our overall research plan to identify hypernymy in as many ways as possible, to allow a multi-faceted disambiguation of hypernym relations in *Roget's* Thesaurus.

#### 3.1 Identifying Hypernyms in Existing Ontologies

Existing lexical resources are an obvious source of lexical relations. We worked both with *WordNet* [2] and *OpenCyc* [13]. The relation between a pair of words in *Roget's* is labeled as a hypernym if the two words have a hypernym/hyponym relationship in *WordNet*. The hypernyms in *WordNet* can be any distance from each other in the hypernym tree. The only requirement is that both hypernym and hyponym appear in the same Paragraph in *Roget's*. We have identified 53,404 relations using *WordNet*.

*OpenCyc* is a freely distributed version of *Cyc*, a large general knowledge base. Although not intended as a lexical ontology, it contains a hierarchy of classes and subclasses, called “genls”. Phrases are also included in *Cyc*, generally rendered as a single word; for example “PlatonicIdea” stands for “platononic idea”. *OpenCyc* contains only a fraction of the relations from the full version of *Cyc*, but we still identified 1,608 relations.

#### 3.2 Identifying Hypernyms in Dictionaries

A second source of hypernym/hyponym pairs are machine-readable dictionaries, among them LDOCE [14], often used in the past to find relations in text. We identify hypernym relations in LDOCE using patterns similar to the two presented by Nakamura and Nagao [10].

Nakamura and Nagao [10] have shown these patterns to work well for LDOCE. We also tried to apply them to Wiktionary [15]. This is somewhat more difficult.

Wiktionary, unlike LDOCE, is not built by professionals (not systematically), so patterns frequent in LDOCE may not appear as frequently in Wiktionary. In the end, we found 5,153 hypernyms in LDOCE and 4,483 in Wiktionary that appear in *Roget's*.

### 3.3 Identifying Hypernyms in a Large Corpus

We identify hypernym relations from text, using the six patterns proposed by Hearst [5] on two different resources: the BNC [16] and the Waterloo MultiText System [17]. The BNC already labels each word/phrase with a part-of-speech tag, which is convenient for implementing Hearst's patterns. We used them across all the BNC and discovered almost 30,000 relations, but only 1332 relations appeared in the same *Roget's* paragraph.

The WMT corpus [17] contains half a terabyte of queryable Web data. We ran queries for specific terms in conjunction with Hearst's patterns, for example "such *NP* as *football*" or "Protestant and other *NP*". First we compiled a list of terms that had no hypernyms assigned by any other method we describe in this paper<sup>1</sup>. The list contained 26,430 unique terms. Of the 26,430 unique words searched for, 15,443 had at least one phrase retrieved using this method. Once the phrases have been extracted, they were tagged using Brill's tagger. Since the WMT corpus does not count punctuation in its patterns, many of the extracted sentences could not match Hearst's patterns due to incorrect or irregular punctuation. For 11,392 relations both terms appear in the same *Roget's* Paragraph.

### 3.4 Labeling a Hypernym Network in *Roget's*

The methods we have presented identified 68,717 unique hypernyms, appearing 92,675 times in *Roget's* Thesaurus. The difference is due to the fact that some hypernyms appear in more than one paragraph. Once this has been done, we removed all cycles and redundant hypernym links. A cycle is a series of hypernym links where a term can eventually become its own hypernym, of the form "A is a B is a ... is a C is an A". We fix cycles by removing the link that is least likely to be correct. In Section 4.1 we discuss how we determine the accuracy of the hypernyms based on scores assigned by human evaluators. We found 3,756 cycles; the average cycle length was 3.8 links.

Redundant hypernym links appear when there is a series of relations "A is a B is a ... is a C" and also a link "A is a C". The relation "A is a C" is not incorrect, but it is redundant. We dropped 30,068 redundant hypernym links. After these two fixes, 58,851 non-unique hypernym relations remained.

## 4 Evaluation

### 4.1 Manual Evaluation of Hypernyms

We asked five evaluators, fluent in English, to evaluate the automatically acquired relations as true or false hypernymy. We sampled 200 pairs from each of the six resources.

---

<sup>1</sup> Two other methods of inferring hypernyms from synonyms were attempted, with poor results. They are not included.

The evaluators did not know from which resource the samples came. One evaluation was incomplete. Table 1 shows the scores from each evaluator (R1..R5) as well as the average score Av and Fleiss' Kappa K [18].

**Table 1.** Raters R1-R5: kappa, precision, recall, F-measure

	R1	R2	R3	R4	R5	Av	K	Total	P	R	F
<b>BNC</b>	.68	.61	.75	.67	.62	.66	.44	1,332	.663	.017	.034
<b>CYC</b>	.95	.78	.86	.86	.89	.87	.38	1,608	.865	.021	.041
<b>LDOCE</b>	.85	.59	.82	.73	.93	.78	.27	5,153	.782	.067	.123
<b>WMT</b>	.72	.39	.51	.51	.57	.54	.37	11,392	.536	.147	.231
<b>Wiki</b>	.73	.56	.75	-	.87	.73	.17	4,483	.726	.057	.107
<b>WN</b>	.86	.52	.80	-	.77	.74	.11	53,404	.735	.690	.712

WMT was the least accurate resource. This is likely due to our using WMT as a last resort: to find relations for terms/phrases not found by any other method. Such terms may be less frequent or may represent concepts harder to identify.

The kappa scores – see Table 1, column K – were not high, particularly for the hypernyms identified in Wiktionary and *WordNet*. These kappa results are somewhat lower than the score of 0.51 shown in Rydin [7] on a similar problem. The low kappa scores and the fact that *WordNet* scored relatively poorly suggests that people are not always good judges of hypernymy. *WordNet's* low scores may be because some hypernyms links appear to be closer to synonymy than to actual hypernymy or because it has many infrequent words senses, of which evaluators may not have been aware.

It is possible to evaluate each resource using precision and recall – see Table 1. Precision (P) is the accuracy of the resource and recall (R) is the proportion of relations found in that resource. Also shown are the total number of relations found in each resource (Total) and the F-measure (F).

## 4.2 Combining the Hypernyms from the Resources

The sets of hypernyms we have produced had to be combined in a way that promises high accuracy. Table 1 shows average accuracy for each resource. These results can be used to determine new accuracies for hypernyms that come from two or more resources. The counts of co-occurring hypernyms for 1-6 resources are 61581, 5839, 1102, 171, 21 and 3.

We used the accuracy assigned to each hypernym pair to break cycles, as discussed in Section 3.4. Let the probability of a false hypernym classified as true in resource *A* be  $P(A)$  (it is  $1 - \text{Accuracy}$  from Table 1). When a hypernym pair  $x$  appears in just one resource, the probability of error is  $P(x) = P(A)$ . If  $x$  is found in more than one resource,  $P(x)$  is calculated as  $P(x) = P(A) * P(B) * \dots * P(Z)$ . Once we have determined the probability of error for each hypernym pair, we can determine the average error for the entire set of hypernyms. The total average accuracy is 73.1% over all 68,717 hypernym pairs. Due to the low kappa scores (Table 1), this accuracy may not be entirely reliable. With this method, some disambiguated hypernyms have extremely high accuracy. The

hypernyms “drill is a tool”, “crow is a bird” and “cactus is a plant” were found in all 6 resources and had probabilities close to 1.0.

### 4.3 Evaluation Through Applications

The last method of evaluating the disambiguated hypernyms in *Roget's* Thesaurus is to test the enhanced Thesaurus on the same applications on which the original (unenhanced) *Roget's* system was tested. The chosen applications make use of a new semantic similarity function that accounts for hypernyms. We start off with a function presented in Jarmasz and Szpakowicz [1] (as seen in Section 1.1), and adjust it for hypernymy. If the two terms are direct hypernyms/hyponyms of each other, we increase the score by 4. We add 3 if there are two hypernym/hyponym links between the terms, 2 for three links, 1 for four links. A penalty of -1 applies to both words if they have no hypernym/hyponym links; this is done because sometimes the relations between a word and the other words in its Head, Paragraph and even Semicolon Group are not clear. If no hypernym for that term exists in its Head, then it becomes more likely that the Paragraph that contains the word does not really represent its true sense. We chose these values because they add reward/penalty to the original similarity function without completely overwhelming it. All this gives a range of scores -2..20, which we shift up to 0..22 to get only non-negative values.

**Semantic Distance and Correlation with Human Annotators.** We tested the new and old semantic similarity measures on three data sets: Miller & Charles [19], Rubenstein & Goodenough [20] and Finkelstein et al. [21]<sup>2</sup>. We measured the Pearson product-moment correlation coefficient between the numbers given by human judges and those achieved by the two systems. The results appear in Table 2 where we compare the original and enhanced semantic distance function. We considered only nouns for this task. We found that the improvement on Rubenstein & Goodenough [20] and Finkelstein et al. [21] was statistically significant with a P-value  $p < 0.05$  using a Paired Student t-test.

**Table 2.** Results for semantic distances and choosing the correct synonym

Data Set	Orig	Enh	ESL		TOEFL		RDWP		
			Right	Wrong	Right	Wrong	Right	Wrong	
Miller & Charles	0.773	0.836							
Rubenstein & Goodenough	0.781	0.838	38	42	58	58	201	205	
Finkelstein et al.	0.411	0.435	12	8	22	22	99	95	
			Ties	3	0	5	5	23	13

**Synonym Identification Problems.** The same semantic similarity function can also work for the problem of identifying a correct synonym of a word from a group of

<sup>2</sup> The WordSimilarity-353 Test Collection is available at: <http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/wordsim353.html>

candidates. We tried a method similar to that found in Jarmasz and Szpakowicz [1]. We used three data sets for this application: Test Of English as a Foreign Language (TOEFL) [22], English as a Second Language (ESL) [23] and Reader’s Digest Word Power Game (RDWP) [24]. See Table 2 for the results of the original and the enhanced system. We show the number of correct, incorrect and tied answers; ties are *also* counted as incorrect.

**Analogy Identification Problems.** In an analogy identification problem we get words  $W_1, W_2$  and we choose among several other pairs the pair linked by the same relation as  $W_1, W_2$ . In this way, it is a relation disambiguation problem since the relations between the words is not known. We worked with 374 SAT analogy questions [25] where the correct analogy is selected among five possibilities. The focus of our work is on a subset of the 374 SAT analogy problem where we can identify hypernym relations using the enhanced *Roget’s* Thesaurus. For the sake of completeness we do tests on the entire data set, but – since only a fraction of the relations are hypernyms – we cannot expect any improvements to be very large. Let the words in the original pair be  $A$  and  $B$  and in the candidate pair  $C$  and  $D$ . The distance formula is as follows<sup>3</sup>:

$$dist = |semDist(A, B) - semDist(C, D)| + 1 / (semDist(A, C) + semDist(B, D) + 1)$$

Each word in the data set had previously been labeled with its part of speech. The candidate pair with the lowest distance score is chosen as the correct analogy. We can also modify the function by checking for hypernym analogies. If both the original word pair and one of the analogy candidates are linked by hypernymy, we can prefer that candidate. In such “hypernymy matching”, we take into account the number of hypernym links between two terms in the original pair and the potential analogy pair; *dist* is altered by this formula:

$$distAlt = dist - (k - |hypernymDist(A, B) - hypernymDist(C, D)|)$$

Here,  $k$  is a constant. Ideally  $k$  should be a suitably high number, so that pairs of words that are both related by hypernymy are favoured above pairs that are not both related by hypernymy. In our case, the selected analogy pair rarely had a *dist* score greater than 8, so we chose  $k = 8$ . We tested four different variations on this algorithm – see Table 3 – on hypernym questions, and all questions. There are 24 cases where the original word pair can be matched by hypernyms. Of these 24 pairs, six more were found to be correct using this new system. All three enhanced systems show considerable improvement over simply using the original semantic distance function without any sort of hypernym matching. All three enhanced methods were found to be statistically significant with a P-value  $p < 0.05$  using a Paired Student t-test on the 24-case subset, though not on the full SAT analogy dataset.

<sup>3</sup> The formula comes from Jarmasz, M., Nastase, V., Szpakowicz, S.: *Roget’s* Thesaurus as an Electronic Lexical Knowledge Base for Natural Language Processing (submitted to *Language Resources and Evaluation*).

**Table 3.** Results for choosing the correct analogy from a set of candidates

System	Right	Wrong	Ties	Omit
Hypernyms Only				
Original	7	15	2	
Original with Hypernym Matching	13	9	2	
Enhanced	13	10	1	
Enhanced with Hypernym Matching	14	9	1	
All Data				
Original	124	226	14	10
Original with Hypernym Matching	130	220	14	10
Enhanced	129	231	4	10
Enhanced with Hypernym Matching	130	230	4	10

## 5 Conclusion

It was difficult to get strong agreement between raters. With kappa scores ranging between .11 and .44, the rater agreement was not high at all. When we average the accuracy for each of the hypernyms identified with each resource (as determined by averaging the results from the human annotators), the hypernyms disambiguated in *Roget's* Thesaurus are 73% accurate. The accuracy of the hypernyms identified ranges from nearly 100% to as low as 53%.

The enhanced *Roget's* Thesaurus worked better than the original ELKB for most of the data sets on which it was tested. We found statistically significant improvements for the data in Rubenstein & Goodenough [20] and Finkelstein et al. [21]. Improvements on these two data sets as well as on the Miller & Charles data [19] are fairly substantial given the already high scores obtained using the unenhanced *Roget's* Thesaurus. We also found small improvements for the ESL [23] and RDWP [24] data sets. The TOEFL set [22] did not show any improvement, but it did no worse either.

We also found some improvement in solving the SAT Analogy questions [25]. Using the improved semantic distance function did improve the results for answering SAT questions, but the best improvements came from matching hypernym analogies to hypernym solutions. This system could be more effective if more hypernym relations as well as other relations were disambiguated in *Roget's*. The problem of solving analogy questions is not an easy one for people or machines. The most successful system that we are aware of is 56% accurate on the same SAT data set [26], while the average college-bound high-school student gets about 57% accuracy.

## Acknowledgment

Our research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the University of Ottawa. We would also like to thank Dr. Diana Inkpen, Anna Kazantseva, Darren Kipp and Dr. Vivi Nastase for reading this paper and providing many useful comments.



## References

1. Jarmasz, M., Szpakowicz, S.: Roget's thesaurus and semantic similarity. In: Proc. Conference on Recent Advances in Natural Language Processing (RANLP 2003), pp. 212–219 (2003)
2. Fellbaum, C. (ed.): WordNet – An electronic lexical database. MIT Press, Cambridge, Massachusetts, London, and England (1998)
3. Kirkpatrick, B. (ed.): Roget's Thesaurus of English Words and Phrases. Penguin, Harmondsworth, Middlesex, England (1987)
4. Jarmasz, M., Szpakowicz, S.: The design and implementation of an electronic lexical knowledge base. In: Proc. 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence (AI 2001), pp. 325–334 (2001)
5. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proc. 14th Conference on Computational linguistics, pp. 539–545 (1992)
6. Caraballo, S.A., Charniak, E.: Determining the specificity of nouns from text. In: Proceedings the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing (EMNLP) and Very Large Corpora (VLC), pp. 63–70 (1999)
7. Rydin, S.: Building a hyponymy lexicon with hierarchical structure. In: Proc. SIGLEX Workshop on Unsupervised Lexical Acquisition, ACL'02, pp. 26–33 (2002)
8. Cederberg, S., Widdows, D.: Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In: Proc. Seventh Conference on Natural Language Learning at HLT-NAACL 2003, pp. 111–118 (2003)
9. Snow, R., Jurafsky, D., Ng, A.Y.: Learning syntactic patterns for automatic hypernym discovery. In: Saul, L.K., Weiss, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems, vol. 17, pp. 1297–1304. MIT Press, Cambridge, MA (2005)
10. Nakamura, J., Nagao, M.: Extraction of semantic information from an ordinary english dictionary and its evaluation. In: Proc 12th Conference on Computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics, pp. 459–464 (1988)
11. Pantel, P., Pennacchiotti, M.: Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In: Proc. 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Sydney, Australia, Association for Computational Linguistics (July 2006), pp. 113–120 (2006)
12. Pantel, P., Ravichandran, D.: Automatically labeling semantic classes. In: Proc. 2004 Human Language Technology Conference (HLT-NAACL-04), pp. 321–328 (2004)
13. Lenat, D.B.: Cyc: A large-scale investment in knowledge infrastructure. Communications of the ACM 38(11) (November 1995)
14. Procter, P.: Longman Dictionary of Contemporary English. Longman Group Ltd. (1978)
15. Wiktionary: Main page - wiktionary (2006), [http://en.wiktionary.org/wiki/Main\\_Page/](http://en.wiktionary.org/wiki/Main_Page/)
16. Burnard, L.: Reference guide for the british national corpus (world edition) (2000)
17. Clarke, C.L.A., Terra, E.L.: Passage retrieval vs. document retrieval for factoid question answering. In: SIGIR '03: Proc. 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 427–428. ACM Press, New York (2003)
18. Fleiss, J.L.: Statistical Methods for Rates and Proportions, 2nd edn. John Wiley & Sons, New York (1981)
19. Miller, G.A., Charles, W.G.: Contextual correlates of semantic similarity. Language and Cognitive Process 6(1), 1–28 (1991)
20. Rubenstein, H., Goodenough, J.B.: Contextual correlates of synonymy. Communication of the ACM 8(10), 627–633 (1965)

21. Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E.: Placing search in context: the concept revisited. In: WWW '01: Proc. 10th International Conference on World Wide Web, pp. 406–414. ACM Press, New York (2001)
22. Landauer, T., Dumais, S.: A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review* 104, 211–240 (1997)
23. Turney, P.: Mining the web for synonyms: Pmi-ir versus lsa on toefl. In: Flach, P.A., De Raedt, L. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 491–502. Springer, Heidelberg (2001)
24. Lewis, M. (ed.): Readers Digest, 158(932, 934, 935, 936, 937, 938, 939, 940), 159(944, 948). Readers Digest Magazines Canada Limited (2000-2001)
25. Turney, P., Littman, M., Bigham, J., Shnayder, V.: Combining independent modules to solve multiple-choice synonym and analogy problems. In: Proceedings International Conference on Recent Advances in Natural Language Processing (RANLP-03), pp. 482–489 (2003)
26. Turney, P.: Similarity of semantic relations. *Computational Linguistics* 32(3), 379–416 (2006)