A Supervised Method of Feature Weighting for Measuring Semantic Relatedness

Alistair Kennedy[1] and Stan Szpakowicz[1,2]

[1] SITE, University of Ottawa, Ottawa, Ontario, Canada
{akennedy,szpak}@site.uottawa.ca
[2] Institute of Computer Science
Polish Academy of Sciences, Warsaw, Poland

Abstract. The clustering of related words is crucial for a variety of Natural Language Processing applications. Many known techniques of word clustering use the context of a word to determine its meaning. Words which frequently appear in similar contexts are assumed to have similar meanings. Word clustering usually applies the weighting of contexts, based on some measure of their importance. One of the most popular measures is Pointwise Mutual Information. It 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. We present a method of supervised feature weighting. It identifies contexts shared by pairs of words known to be semantically related or unrelated, and then uses Pointwise Mutual Information to weight these contexts on how well they indicate closely related words. We use *Roget's Thesaurus* as a source of training and evaluation data. This work is as a step towards adding new terms to *Roget's Thesaurus* automatically, and doing so with high confidence.

1 Introduction

Pointwise Mutual Information (PMI) is a measure of association between two values of two random variables. PMI has been applied to a variety of Natural Language Processing (NLP) tasks, and shown to work well when identifying contexts indicative of a given word. In effect, PMI can be used to give higher weights to contexts in which a word occurs frequently, but other words appear rarely, while giving lower weight to contexts with distributions closer to random. Finding these weights requires no actual training data, so it is essentially an unsupervised method of context weighting, an observation also made in [1]. In our paper we show how to incorporate supervision into the process of context weighting. We learn appropriate weights for the contexts from known sets of related and unrelated words extracted from a thesaurus. PMI is then calculated for each context: we measure the association between pairs of words which appear in that context and pairs of words which are known to be semantically related. The PMI scores can then be used to apply a weight to the contexts in which a word is found. This is done by building a *word-context* matrix which records the counts

C. Butz and P. Lingras (Eds.): Canadian AI 2011, LNAI 6657, pp. 222–233, 2011.

of how many times each word appears in each context. By applying our weighting technique to this matrix, we are effectively training a measure of semantic relatedness (MSR). In our experiments, unsupervised PMI measures association between a context and a word, while supervised PMI measures association between a context and synonymy. We also perform experiments combining these supervised and unsupervised methods of learning semantic relatedness between word pairs.

Our system uses data from two versions of *Roget's Thesaurus*, from 1911 and from 1987, for our supervised context weighting method. We also compare the two versions of *Roget's* and determine how its age and size affect it as a source of training data. We use *SuperMatrix* [2], a system which implements a variety of MSRs. Specifically, we use its Cosine similarity and PMI MSRs. The corpus we use for building our *word-context* matrix is Wikipedia.

Motivation

This work is designed to be a step towards automatically updating *Roget's Thesaurus* through identifying semantically related words and clustering them. Our goal is not to create a new thesaurus from scratch but rather to update an existing one. We can therefore try to use the existing thesaurus as a tool for learning how words are related, which in turn can help update *Roget's*. Rather than relying on unsupervised word similarity metrics, we can use *Roget's Thesaurus* to train potentially superior word similarity metrics. This has been partially inspired by [3], where machine learning is used to learn from a corpus words related by hypernymy. Training on known hypernym and non-hypernym pairs in *WordNet* [4] allows the system to learn to identify hypernyms for adding to *WordNet*. *Roget's* is structured quite differently from *WordNet*, so the technique of [3] is not appropriate here, but we adopt the "bootstrapping" idea of using a lexical resource to aid in its own expansion.

2 Related Work

A variety of corpus-based Measures of Semantic Relatedness (MSRs) have been developed by NLP researchers – see [5] for an in-depth review. Corpus-based MSRs generally work by representing word w as a vector of contexts in which w appears. The context can be as broad as the document where w appears [6], or as specific as one word, for example in a verb-object relationship with w [7].

Contexts are most often determined using a dependency parser to extract from the text triples $\langle w, r, w' \rangle$, where word w is related to another word w' by relationship r. The context of w is then the pair $\langle r, w' \rangle$. This technique has been widely applied [8–11].

There have been attempts to incorporate some supervision into the process of learning semantic distance. In [12], a function consisting of weighted combinations of precision and recall of contexts is proposed for measuring semantic relatedness. In this function there are two thresholds which the authors optimize

using a set of training instances. Many variations on their measure were evaluated on the task of predicting how closely word clusters match that of a thesaurus (as we do), and on pseudo-word-sense-disambiguation. This involves minimal supervision: only two thresholds are learned.

There also is related work on learning weights for short document similarity. In [13, 14] a method of learning weights in a *word-document* matrix was proposed. The authors weighted terms to learn document similarity rather than weighting contexts to learn word similarity. The method was to minimize a loss function rather than to apply PMI. They compared their system against TF.IDF weighting of documents. The documents they used were actually queries and the task was to identify advertisements relevant to a given query.

[15] presents another related project. A combination of supervised and unsupervised learning determines whether one verb can be a paraphrase of another. Unsupervised learning is used to bootstrap instances where one verb can be replaced by another. These bootstrapped examples are then used to train a classifier which can tell in what contexts one word can replace another.

A supervised method of learning synonyms in [1] is probably the work most closely related to ours. A variety of methods, both distributional and pattern-based, for identifying synonymy is followed by machine learning to combine these methods. Such combination was found to give improvement over individual methods. We do not use supervision to combine methods of identifying synonyms but rather to determine the weights for a measure of semantic relatedness.

PMI itself has been widely used in NLP. In [16], PMI is used to learn word sentiment by measuring the association between a phrase and other words known to be positive or negative. PMI has also been applied to named entity extraction from text [17] and query classification into types [18]. In [19], PMI is used in an unsupervised manner to assign weights to a *word-context* matrix. This process is further described in Section 3.

3 Unsupervised Use of PMI for Measuring Semantic Relatedness

We use PMI for both supervised and unsupervised learning of context weights. In this section we describe how PMI is used in an unsupervised way. PMI is actually a measure of association between two events, x and y:

$$PMI(x, y) = log\left(\frac{P(x, y)}{P(x) * P(y)}\right) \tag{1}$$

When those two events are a particular word and a particular context, we can measure association between them and use this as a weighting scheme for measuring semantic distance [19]. This is what is calculated when using PMI for unsupervised *term-context* matrix weighting. To create the *term-context* matrix we used a tool called *SuperMatrix*.

3.1 *SuperMatrix*

SuperMatrix [2] is a tool which has implemented a large variety of MSRs on a *word-context* matrix. These include other variations on PMI [20] and Lin's measure [8], and measures proposed in [12]. A number of variations on these measures and many others, all referred to as RankWeight Function (RWF) [21, 22] have been implemented and are shown to enhance many of those measures. RWF is interesting as it applies one context weighting function on top of another. Likewise, we will apply different weighting methods on top of each other when we combine supervised and unsupervised context weighting.

To use *SuperMatrix*, we give it a single query word q and ask for it to return the set of 100 words $w_1..w_{100}$ most closely related to q.[1]

To construct a *word-context* matrix to run the *SuperMatrix* MSRs, we applied the same methods as [8]. We parsed with Minipar [23] a corpus comprised of about 70% of Wikipedia.[2] The parsing results supply dependency triples $\langle w, r, w' \rangle$. We split these triples into two parts: a word w and a pair $\langle r, w' \rangle$ – the context in which w is found. Examples of triples are $\langle time, mod, unlimited \rangle$ and $\langle time, conj, motion \rangle$, where the word "time" appears in the context with the modifier "unlimited" and in a conjunction with "motion".

The *word-context* matrix is constructed from these dependency triples. Each row corresponds to a word w, each column – to one of the contexts, C. That cell of the matrix records $count(w, C)$: how many times w is found in C. As we learn either supervised or unsupervised weights, we change the values in this matrix from straight counts to more appropriate weights. Each row in this matrix is essentially a vector representing a word. The distance between two words is the distance between their vectors.

To reduce noise, only words appearing 50 or more times and contexts appearing 5 or more times are included. This gives us a total of 32743 words and 321152 contexts. The average word appears in approximately 480 unique contexts, while each context appears as a feature in around 50 words. We only used nouns in our experiments.

3.2 Applying Unsupervised PMI

A PMI score determines to what extent a word and a context appear together beyond random chance. In this case we have the probabilities $P(x)$ of seeing the word, $P(y)$ of seeing the context and $P(x, y)$ of seeing both together. This is calculated for all contexts in all word vectors. The actual distance between two words a and b is the distance between the vectors of contexts for those words, A and B respectively. One of the most common means of measuring distance between vectors – and indeed the measure we apply – is cosine similarity:

$$cos(A, B) = \frac{A \bullet B}{\|A\|\|B\|} \tag{2}$$

[1] Scores for each word, in the range $\langle 0..1 \rangle$, are provided, but we only need rank.

[2] That was a dump of August 2010. 70% was the most data we could process on a computer with 4GB of RAM.

Vectors which appear closer together are assumed to have much more similar meaning while vectors that appear farther apart are assumed to have less related meanings. Our two unsupervised MSRs will be plain cosine similarity and PMI weighting with cosine similarity.

4 Supervised Learning of Context Weights

In this section we describe how a weight for each context is learned. For this we need training data, we turn to *Roget's Thesaurus* to provide us with lists of known related and unrelated words.

4.1 *Roget's Thesaurus*

Roget's Thesaurus is a nine-level hierarchical thesaurus. The levels, from top to bottom, are *Class* → *Section* → *Sub-Section* → *Head Group* → *Head* → *Part of Speech* → *Paragraph* → *Semicolon Group* → *Words/Phrases*. Earliest published versions of *Roget's* come from the 1850s, but it has been constantly under revision: new editions are released every few years. We will use two version of *Roget's*. *Open Roget's* [24] is a publicly available Java implementation intended for use in NLP research, built on *Roget's* data from 1911.[3] The second version is proprietary, based on data from the 1987 edition [25]. Generally we prefer to work with public-domain resources. Still, the 1987 *Roget's Thesaurus* gives us an opportunity to see how a newer and larger resource compares to an older and smaller one.

Roget's contains a variety of words and phrases divided into four main parts of speech: Nouns, Verbs, Adjectives and Adverbs. In our experiments we will only work with Nouns. The main concepts in *Roget's* are often considered to be represented by the Heads, of which there are usually about 1000. The division into parts of speech occurs between the Head and the Paragraph, so that each main concept (Head) contains words in different parts of speech. The smallest grouping in *Roget's* is the Semicolon Group (SG), while the next smallest is the Paragraph. SGs group together near-synonyms, while Paragraphs tend to contain a little more loosely related words. An example of some of the Noun SGs and Paragraphs from the Head for "Language" can be seen in Figure 1. Each SG is delimited by a semicolon while Paragraphs start with an italicized word/phrase and end in a period.

Our evaluation requires information from the SG and Paragraphs in *Roget's*. Table 1 shows the statistics of those groupings: the counts of Noun Paragraphs, SGs, their average sizes in words, and the total count of all Nouns. The latter includes duplicates when a noun appears in two or more SGs. A phrase counts as a single word, although the individual words inside it could be used as well. The 1911 *Roget's* has more paragraphs, but the 1987 version has more SGs, more words and a higher average number of words in each grouping. The 1987 *Thesaurus* should be better for evaluation: it simply has more labeled data.

[3] `rogets.site.uottawa.ca`

language; phraseology; speech; tongue, lingo, vernacular; mother tongue, vulgar tongue, native tongue; household words; King's English, Queen's English; dialect.
confusion of tongues, Babel, pasigraphie; pantomime; onomatopoeia; betacism, mimmation, myatism, nunnation; pasigraphy.
lexicology, philology, glossology, glottology; linguistics, chrestomathy; paleology, paleography; comparative grammar.

Fig. 1. Excerpt from the Head for "Language" in the 1911 *Roget's Thesaurus*

Table 1. Counts of Semicolon Groups and Paragraphs, their average sizes, and all Nouns in *Roget's Thesaurus*

Year	Para Count	Words per Para	SG Count	Words per SG	Noun Count
1911	4495	10.3	19215	2.4	46308
1987	2884	39.7	31174	3.7	114473

4.2 Supervised Weighting

We want to measure the association between pairs of words appearing in a context and a pair of words appearing in the same SG. For each context C, all the words $w_1..w_n$ which appear in C are collected and all pairs of these words are recorded. C is a pair $\langle r, w' \rangle$, while each word w_i in $w_1..w_n$ appears in the triple $\langle w_i, r, w' \rangle$ in the parsed Wikipedia. We then find in *Roget's* all words in the same SG as $w_i \in \langle w_1..w_n \rangle$, and record these pairs. Only the words also found in our *word-context* matrix are included in these counts. These groups of word pairs can be treated as events for which we measure the Pointwise Mutual Information, effectively giving the context C a score. Words which appear in our set of 500 test cases are not included when learning the weights of the contexts. To calculate the PMI, we count the following pairs of words w_i, w_j (C is a context):

- w_i and w_j are in the same SG and share C [True Positives (tp)];
- w_i and w_j are in different SGs and share C [False Positives (fp)];
- w_i and w_j are in the same SG and only one of them appears in C [False Negatives (fn)];
- w_i and w_j are in different SGs and only one of them appears in C [True Negatives (tn)].

We define the probability of event x as $P(x) = x/(tp+tn+fn+fp)$. Essentially we build a confusion matrix and from it calculate the probabilities. Next, we calculate the PMI for context C, effectively giving a score to this context.

$$score(C) = log\left(\frac{P(tp)}{P(tp+fp) * P(tp+fn)}\right) \tag{3}$$

This is repeated for every context in our *word-context* matrix. Once all the scores have been generated, we can use them to re-weight our *word-context* matrix. For

every word w_i which appears in a given context C, its count $count(w_i, C)$ is multiplied by $score(C)$.

Calculating this number for all contexts is not trouble-free. For one, not all contexts will appear in the training data. To avoid this, we normalize every $score(C)$ calculated in Equation 3 so that the average $score(C)$ is 1; next, we assume that any unseen contexts also have a weight of 1; finally, we multiply the count of context C by $score(C)$ for every word in which C appears. Another problem is that PMI may give a negative score when the two events are less likely to occur together than by chance. In such situations we set $score(C)$ to zero. Another problem is that often the supervised PMI is calculated with a fairly small number of true positives and false negatives, so it may be difficult to get a very reliable score. The unsupervised PMI matrix weighting, on the other hand, will use the distributions of a word and context across the whole matrix, so often will have more data to work with. It may, then, be optimistic to think that supervised PMI will on its own outperform unsupervised PMI. The more interesting experiments will be to see the effects of combining supervised and unsupervised PMI MSRs.

4.3 Experiment Setup

The problem on which we evaluate our technique is that of ranking closely related words. We select a random set of 500 words found in our *SuperMatrix* matrix and both in the 1911 and 1987 *Roget's Thesaurus*, from a possible set of 11725. These 500 words were not used for matrix weighting, described in Section 4.2. For each of these words we use our MSRs to generate a ranked list of the 100 most closely related words in our matrix. These lists are evaluated for accuracy at various levels of recall using *Roget's Thesaurus* as a gold standard. Specifically we measure the accuracy at the top 1, 5, 10, 20, 40 and 80 words. We take words from a list of the top 100 but not all of these 100 words will appear in *Roget's*. That is why there will be cases in which we cannot find all 40 or 80 words to perform our evaluation. In such cases we simply perform our evaluation on all the words we can use from that list of 100.

As shown in Table 1, the newer and larger 1987 version contains more words known to be semantically related than the 1911 version, so we will only use it for evaluation. We measure accuracy at identifying words in the same SG and the same Paragraph. This is done because, when adding new words to *Roget's*, one may want to take advantage of both the closely related words (SG) and more loosely related words (Paragraph).

In our evaluation we run six different MSRs. We use unsupervised cosine similarity and an unsupervised PMI MSRs as low and high baselines. We also test cosine similarity when context weights are learned using both the 1987 and 1911 *Roget's Thesaurus*. These MSRs are denoted 1987-Cosine and 1911-Cosine. They can be compared to the unsupervised PMI MSR. Finally we attempt to combine the supervised and unsupervised matrix weighting. This is done by first applying the weighting learned through supervision to the *word-context* matrix and then using the unsupervised PMI MSR on that matrix, once again for both

versions of *Roget's*. These MSRs are denoted 1987-PMI and 1911-PMI. Although this may not seem intuitive, it is not so different from the RWF measures, in that two ranking methods are combined. Sample lists generated with two of these measures, 1911-Cosine and PMI, appear in Figure 2.

1911-Cosine – backbencher (0.715), spending (0.657), bureaucracy (0.645), funding (0.619), agency (0.616)
PMI – incentive (0.200), funding (0.192), tax (0.187), tariff (0.180), payment (0.176)

Fig. 2. The top 5 words related to "Subsidy", with their similarity score using the supervised 1911-Cosine MSR and unsupervised PMI

5 Experiment Results

We evaluate our new supervised MSRs as well as the unsupervised MSRs on two kinds of problems. In one, we evaluate the ranked list by calculating its accuracy in finding words in the same SG. The second evaluation is done by determining accuracy at finding words in the same Paragraph.

5.1 Ranking Words by Semicolon Group

We count the number of words found to be in the same SG and those known to be found in different SGs in *Roget's Thesaurus*. From this we calculate the accuracy of each MSR for the top 1, 5, 10, 20, 40 and 80 related words – see Table 2. In evaluating our results, we broke the data into 25 sets of 20 lists and performed Student's t-test to measure statistical significance at $p < 0.05$. The numbers are in bold when a supervised MSR shows a statistically significant improvement over its unsupervised counterpart.

Table 2. Evaluation results for identifying related words in the same Semicolon Groups

Measure	Top 1	Top 5	Top 10	Top 20	Top 40	Top 80
Cosine	.110	.070	.052	.039	.031	.024
PMI	.368	.243	.188	.136	.100	.072
1987-Cosine	**.146**	**.092**	**.071**	**.055**	**.042**	**.034**
1987-PMI	.378	.240	.187	.136	**.101**	.073
1911-Cosine	**.146**	**.097**	**.073**	**.055**	**.042**	**.034**
1911-PMI	.372	.242	.189	**.138**	.100	.073

Our lower baseline MSR – cosine similarity – does quite poorly. In comparison, 1987-Cosine and 1911-Cosine gives a relative improvement of 30-40%. Supervised learning of context weights using PMI improved the Cosine similarity MSR by a statistically significant margin in all cases. Surprisingly, in a number of cases 1911-Cosine performs slightly better than 1987-Cosine. Figure 2 may suggest

why supervised PMI did worse than unsupervised PMI. The latter tended to retrieve closer synonyms, while the former selected many other related words.

Supervised matrix weighting with PMI (1911-Cosine and 1987-Cosine) did not work as well as unsupervised matrix weighting with PMI. As noted in Section 4 this is not entirely unexpected. Combining the supervised and unsupervised PMI weighted methods does in some cases show an advantage. 1987-PMI and 1911-PMI showed a statistically significant improvement only when the top 40 and 20 words were counted respectively. That said, in a few cases combining these measures actually hurt results, although never in a statistically significant manner; most often results improved slightly. It is easier to show a change to be statistically significant as more related words are considered, because it provides a more reliable accuracy. This is tested further where we perform evaluation on Paragraphs rather than on SGs.

5.2 Ranking Words by Paragraph

The experiments from Section 5.1 are repeated on Paragraphs – see Table 3. Obviously accuracy at all levels of recall is higher in this evaluation, because there are far more related words in the same Paragraph than in the same SG. Another interesting observation is that the improvement from combining supervised and unsupervised PMI matrix weighting was statistically significant much more often. 1987-PMI showed a statistically significant improvement over PMI when the top 20 or more closest words were used in evaluation. For 1911-PMI the improvement was statistically significant for the top 10 or more closest words. We found improvements of up to 3% when mixing the supervised and unsupervised matrix weighting.

Table 3. Evaluation results for identifying related words in the same Paragraphs

Measure	Top 1	Top 5	Top 10	Top 20	Top 40	Top 80
Cosine	.256	.206	.173	.148	.127	.110
PMI	.624	.524	.466	.401	.345	.287
1987-Cosine	**.298**	**.240**	**.208**	**.180**	**.157**	**.138**
1987-PMI	.644	.523	.470	**.406**	**.349**	**.291**
1911-Cosine	**.296**	**.240**	**.209**	**.182**	**.160**	**.141**
1911-PMI	.640	.533	**.478**	**.416**	**.352**	**.295**

Once again evaluation on the 1911 *Roget's* often performed better than on the 1987 version. It is easier to show statistically significant improvements for Paragraphs than for SGs, because the number of positive candidates grows higher. The data in Table 1 suggest that a word may only have a few other words in the same SG with it, while it will often have dozens of words in the same Paragraph. As a result, when we perform a t-test, each fold contains many more positive examples and so gives better estimate of how much incorporating supervised weighting actually improves these MSRs.

5.3 Possible New Word Senses

We have not taken into account the possibility that new or missing senses of words are being discovered. If we look at the highest-ranked word in each list of candidates, we often find that the word appears to be closely related, but sometimes *Roget's* labels them as not belonging in that Paragraph or SG. The following are a few of the more obvious examples of closely related words which did not appear in the same Paragraph: *invader – invasion*; *infant – newborns*; *mafia – mob* and *evacuation – airlift*. Although not all the candidates labeled as unrelated may be as closely related as these pairs, it appears clear that the accuracies we find should be considered as lower bounds on the actual accuracy.

6 Analysis and Conclusion

We have clearly shown that supervised weighting of *word-context* matrices is a significant improvement over unweighted cosine similarity. Our method of supervised weighting of *word-context* matrices with PMI was not as effective as unsupervised term weighting with PMI. We found, however, that combining supervised and unsupervised matrix weighting schemes often showed a statistically significant improvement. This was particularly the case when identifying more loosely semantically related words, in the same Paragraph rather than limiting occurrences of related words to the same SG. Never did combining supervised and unsupervised learning actually hurt the results in a statistically significant manner. There are simply are not enough words in the average SGs to prove that incorporating supervised training helps the PMI MSR. This is supported by the fact that when enough data is used – the top 10-20 related words – the evaluation on Paragraphs does show a statistically significant improvement.

One surprise was that often weighting the *word-context* on the 1911 *Roget's Thesaurus* performed slightly better than its counterpart weighted with the 1987 version. This is difficult to explain, but the differences between the two trained systems tended to be quite small. This does suggest that the 1911 version of *Roget's* provides sufficient data for weighting of these contexts despite its smaller size. This is particularly good news, because the 1987 version is not publicly available, while the 1911 version is.

6.1 Future Work

The long-term motivation for this work is automatic updating of *Roget's Thesaurus* with new words. The results we present here suggest that the first step toward that goal has been successful. Next, ranked lists will be used to determine which SGs and Paragraphs are good candidate locations for a word to be added.

We applied two version of *Roget's Thesaurus* for training our system, but it is quite possible to use other resources, including *WordNet*. It is also possible to use functions other than PMI for learning matrix weighting. Likelihood ratio tests are known to work well on rare events and should be considered [26].

Finally, let us note that we have only used our supervised matrix weighting technique to enhance Cosine similarity and PMI MSRs. Many other measures are available via *SuperMatrix*, and there are other resources on which supervised matrix weighting could be applied.

Acknowledgments

Our research is supported by the Natural Sciences and Engineering Research Council of Canada and the University of Ottawa.

References

1. Hagiwara, M., Ogawa, Y., Toyama, K.: Supervised synonym acquisition using distributional features and syntactic patterns. Journal of Natural Language Processing 16, 59–83 (2005)
2. Broda, B., Jaworski, D., Piasecki, M.: Parallel, Massive Processing in SuperMatrix – a General Tool for Distributional Semantic Analysis of Corpus. In: Proceedings of the International Multiconference on Computer Science and Information Technology, pp. 373–379 (2010)
3. Snow, R., Jurafsky, D., Ng, A.Y.: Semantic Taxonomy Induction from Heterogenous Evidence. In: Proceedings of COLING/ACL 2006, Sydney, Australia (2006)
4. Fellbaum, C. (ed.): WordNet: an Electronic Lexical Database. MIT Press, Cambridge (1998)
5. Turney, P.D., Pantel, P.: From Frequency to Meaning: Vector Space Models of Semantics. Journal of Artificial Intelligence Research 37, 141–188 (2010)
6. Crouch, C.J.: A Cluster-Based Approach to Thesaurus Construction. In: SIGIR 1988: Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 309–320. ACM, New York (1988)
7. Ruge, G.: Automatic Detection of Thesaurus relations for Information Retrieval Applications. In: Foundations of Computer Science: Potential - Theory - Cognition, to Wilfried Brauer on the Occasion of his Sixtieth Birthday, pp. 499–506. Springer, London (1997)
8. Lin, D.: Automatic retrieval and Clustering of Similar Words. In: Proceedings of the 17th International Conference on Computational Linguistics, pp. 768–774. Association for Computational Linguistics, Morristown (1998)
9. Curran, J.R., Moens, M.: Improvements in Automatic Thesaurus Extraction. In: Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX), pp. 59–66 (2002)
10. Yang, D., Powers, D.M.: Automatic Thesaurus Construction. In: Dobbie, G., Mans, B. (eds.) Thirty-First Australasian Computer Science Conference (ACSC 2008). CRPIT, vol. 74, pp. 147–156. ACS, Wollongong (2008)
11. Rychlý, P., Kilgarriff, A.: An Efficient Algorithm for Building a Distributional Thesaurus (and other Sketch Engine Developments). In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pp. 41–44. Association for Computational Linguistics, Prague (2007)
12. Weeds, J., Weir, D.: Co-occurrence Retrieval: A Flexible Framework for Lexical Distributional Similarity. Comput. Linguist. 31(4), 439–475 (2005)

13. Yih, W.-t.: Learning term-weighting functions for similarity measures. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, vol. 2, pp. 793–802. Association for Computational Linguistics, Morristown (2009)
14. Hajishirzi, H., Yih, W.-t., Kolcz, A.: Adaptive near-duplicate detection via similarity learning. In: Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, pp. 419–426. ACM, New York (2010)
15. Connor, M., Roth, D.: Context sensitive paraphrasing with a global unsupervised classifier. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 104–115. Springer, Heidelberg (2007)
16. Turney, P., Littman, M.: Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus. Technical report NRC technical report ERB-1094, Institute for Information Technology, National Research Council Canada (2002)
17. Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Unsupervised named-entity extraction from the web: an experimental study. Artif. Intell. 165, 91–134 (2005)
18. Kang, I.H., Kim, G.: Query type classification for web document retrieval. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, SIGIR 2003, pp. 64–71. ACM, New York (2003)
19. Pantel, P.A.: Clustering by Committee. PhD thesis, University of Alberta (2003)
20. Evert, S.: The Statistics of Word Cooccurrences: Word Pairs and Collocations. PhD thesis, Universität Stuttgart (2004)
21. Piasecki, M., Szpakowicz, S., Broda, B.: Automatic Selection of Heterogeneous Syntactic Features in Semantic Similarity of Polish Nouns. In: Matoušek, V., Mautner, P. (eds.) TSD 2007. LNCS (LNAI), vol. 4629, pp. 99–106. Springer, Heidelberg (2007)
22. Broda, B., Derwojedowa, M., Piasecki, M., Szpakowicz, S.: Corpus-based Semantic Relatedness for the Construction of Polish WordNet. In: Calzolari, N., (Conference Chair), Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., Tapias, D. (eds.) Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008). European Language Resources Association (ELRA), Marrakech (2008)
23. Lin, D.: Dependency-Based Evaluation of MINIPAR. In: Proceedings of the Workshop on the Evaluation of Parsing Systems, First International Conference on Language Resources and Evaluation (1998)
24. Kennedy, A., Szpakowicz, S.: Evaluating Roget's Thesauri. In: Proceedings of ACL 2008: HLT, pp. 416–424. Association for Computational Linguistics, Morristown (2008)
25. Kirkpatrick, B. (ed.): Roget's Thesaurus of English Words and Phrases . Longman, Harlow (1987)
26. Dunning, T.: Accurate Methods for the Statistics of Surprise and Coincidence. Computational Linguistics 19(1), 61–74 (1993)