Getting Synonym Candidates from Raw Data in the English Lexical Substitution Task

Diana McCarthy, Lexical Computing Ltd, Brighton, UK
Bill Keller, University of Sussex, Falmer UK
Roberto Navigli, Sapienza University of Rome, Rome, Italy

Distributional similarity provides a technique for obtaining semantically related words from corpus data using automated methods that compare the contexts in which the words appear. Such methods can be useful for producing thesauruses, with application to work in lexicography and computational linguistics. However, the most similar words produced using these methods are not always near synonyms, but may be words in other semantic relationships: antonyms, hyponyms or even looser 'topical' relations. This means that manual post-processing of such automatically produced resources to filter out unwanted words may be necessary before they can be used. This paper evaluates the performance of distributional methods for finding synonyms on the English Lexical Substitution Task, a lexical paraphrasing task where it is necessary to generate candidate synonyms for a target word and then select a suitable substitute on the basis of contextual information. We examine the performance of distributional methods for the first step of generating candidate synonyms and leave the second step of choosing a candidate on the basis of context for future work. A number of automated distributional methods are compared to techniques that make use of manually produced thesauruses. We demonstrate that while the performance of such automatic thesaurus acquisition methods is often below manually produced resources, precision can be greatly increased by using two automatic methods in combination. This approach gives precision results that surpass methods that exploit manually constructed resources for the same task, albeit at the expense of coverage. We conclude that such an approach to increase the precision of automatic methods to find near synonyms could improve the use of distributional methods in lexicography.

1. Introduction

For more than a decade, the problem of assigning senses to specific instances of words in text has been the focus of much research in computational linguistics. This sub field of the discipline is referred to as word sense disambiguation (WSD, see Navigli (2009) for a survey). In WSD, human annotators are given a prescribed set of sense definitions for a number of target words and required to label instances of the words in context according to those definitions. Computer systems are then given the same task and evaluated in terms of how well the sense labels they assign correspond to those of the human annotators, where correspondence is typically quantified in terms of the measures of precision (how many items were correct as a percentage of those attempted) and recall (how many items were correct as a percentage over all those in the test set) (Palmer et al., 2006).

A major issue with this enterprise has been the choice of sense inventory, as it is not clear which inventory will suit which computer application (Palmer et al., 2004). To address this issue, one new initiative for evaluation has been a lexical paraphrasing task where it is left up to the participating systems to determine which dictionary or thesaurus to use for candidate senses and synonyms. This initiative was conducted as the English Lexical Substitution (hereafter Lexsub) Task (McCarthy and Navigli, 2007), which was run as part of SemEval[1], a triennial event in computational linguistics focusing on the evaluation of computer systems on a variety of semantics tasks. There are two aspects to the Lexsub task (McCarthy and Navigli, 2009):

[1] http://nlp.cs.swarthmore.edu/semeval

i) generating a suitable set of candidate synonyms (lexical substitutes) for a target word;

ii) selecting the best lexical substitute from the candidates given the context of the word token.

In this paper, we examine the performance of corpus-based methods for the first step and leave the second step for future work. All participating systems in the Lexsub task used manually constructed thesauruses to find sets of candidate synonyms. While manually constructed thesauruses certainly provide a useful starting point for the task, there is scope for finding synonyms from textual corpora automatically. Fully automatic methods offer the advantage that they can be applied to any language and have applications to lexicography, for example in producing lists of putative synonyms for lexicographers (Kilgarriff et al. 2004). While the association of the candidate synonyms with a specific context is also of interest for lexicographers, we do not deal with that issue in this paper but focus on the task of finding good synonyms.

To generate candidate synonyms, we used Lin's measure of distributional similarity (Lin, 1998), as this outperformed other automatic distributional measures in previous studies (McCarthy and Navigli, 2009) and has been applied successfully in lexicographic tools (Kilgarriff et al., 2004). One of the problems of automatically generated distributional thesauruses is that they include semantically related words that are not good paraphrases (near synonyms). For example, in our distributional thesaurus *video* is listed as the most similar word to *film*, with *movie* as the second most similar. We would like to re-rank these candidate substitutes for *film* so that *movie* comes first because this is a better synonym, from intuition, inspection of WordNet and the Lexsub data.[2] In the work reported here, we examined ways of re-ranking the synonym candidates. The re-ranking methods did not improve overall performance on the Lexsub task compared to Lin’s distributional similarity method, but by combining different methods we were able to increase precision at the expense of recall.

2. Background

2.1. The English Lexical Substitution Task (Lexsub)

The Lexsub task was designed to evaluate language processing systems which can

1) generate synonyms and
2) match these synonyms to a context.

A total of 2010 sentences, each containing one of a sample of 201 target words (nouns, verbs, adjectives and adverbs), were extracted from the English Internet Corpus (Sharroff, 2006) so that the data included exactly 10 sentences for each word. Five human annotators were asked to use their judgement to supply a substitute (near synonym) for each target word in context. The annotators were allowed to provide up to three such synonyms if all were judged equally suitable and they could answer NIL if they were unable to find a suitable synonym. The data was divided into a development set (30 words, 300 sentences) and a test set (171 words, 1710 sentences). The resulting data allows for a view of word meaning based on usage rather than discrete senses (Erk et al., 2009) because there is never just one substitute that will fit a given instance (the context is a sentence in this task). Sentences containing the same target word will tend

[2] Available at http://www.dianamccarthy.co.uk/task10index.html

to overlap, to a greater or lesser extent, in terms of the substitutes that are possible in the given context. In figure 1 we show the substitutes from the human annotators for the verb *cry*.

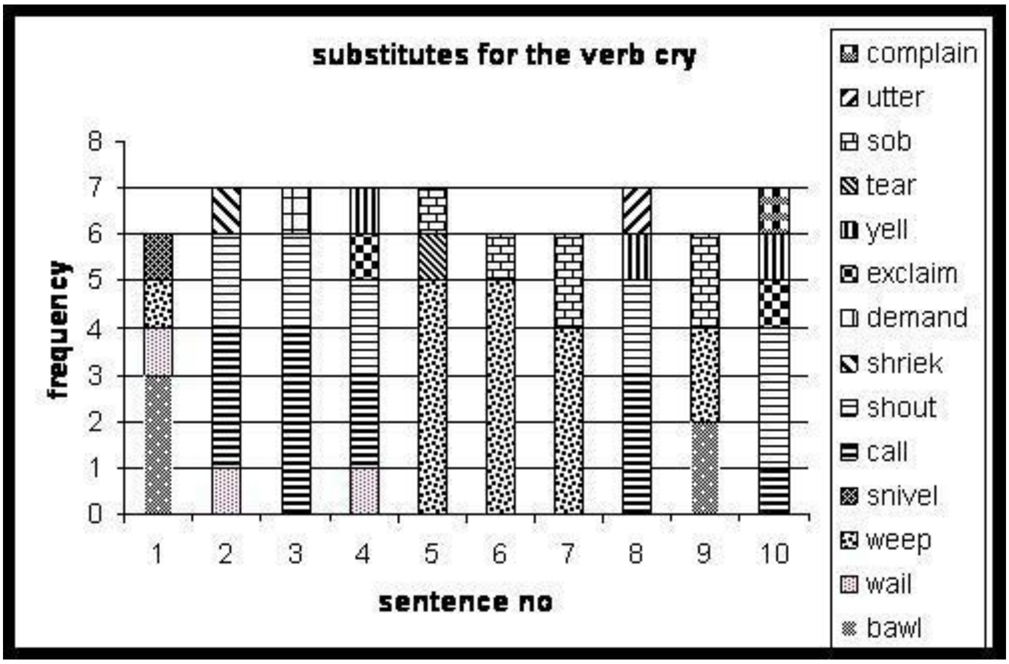

Figure 1. Bar chart showing frequency of substitutes for *cry* in the Lexsub gold standard.

The systems could supply one or more substitutes for a given target word in context. To evaluate the systems, the ‘best’ score[3] is calculated using the average annotator response frequency of substitutes over all target instances[4] (recall) or over just those where the system attempted an answer (precision). Full details of the scoring method can be found in McCarthy and Navigli (2009). It is important to note that for the ‘best’ score, the theoretical upper bound (highest possible score) is 45.76. This is because the credit awarded for each identified substitute is divided by the total number of substitutes for that target word in that sentence, thereby increasing the score where there is greater consensus between annotators. The average pair-wise agreement between annotators was 27.7, showing that even humans do not achieve performance at the theoretical upper bound. This is a hard task because of the inherent variability in finding a synonym in a given context.

In the SemEval evaluation exercise, all participating systems used manually constructed thesauruses. The best participating system had a recall[5] of 12.90. A baseline system using WordNet (Fellbaum 1998), but disregarding the local context of the target word, achieved a recall of 9.95. Against this, the best performing method using only corpus data and distributional methods achieved a recall score of 8.53.

2.2. Distributional Thesauruses

Automatic thesaurus generation is the use of computational methods for discovering semantic relationships between words. It is a challenging problem in natural language learning that has received considerable attention in recent years. A widely-used

[3] There is an alternative ‘out of ten’ score which allows up to 10 answers to be taken into account, but we do not make use of that in this paper.

[4] A target instance is one of the target words in the context of a sentence.

[5] Note that, as every item was attempted, precision is also 12.90.

technique is to apply distributional similarity methods to bootstrap semantic relationships from large, general corpora (e.g. Grefenstette, 1994; Lin, 1998; Curran and Moens, 2002; Weeds, 2003; Rychlý and Kilgarriff, 2007). Underlying this approach is the intuition that two words are similar if they appear in similar distributional contexts. Context may be represented in different ways depending on what counts as a contextual feature. For example, two words might share a contextual feature if they occur in the same document, or the same sentence, or the same grammatical dependency relation (e.g. as the nominal subject or object of a particular verb).

In practice it is usual to take grammatical dependency relations as contextual features. This choice is motivated by the distributional hypothesis of Harris (1968), which predicts that words sharing a large number of grammatical dependency relations should have similar or related meanings. Experimental evidence shows that the use of grammatical relations yields ‘tighter’ thesauruses, in which words are related by paradigmatic semantic relations (synonymy, antonymy, hyponomy) rather than ‘looser’, topical relations. The distributional thesaurus developed for the work described in this paper was based on the written portion of the British National Corpus (Leech, 1992), which was first parsed using the RASP dependency parser (Briscoe & Carroll 2002) to extract grammatical relations. We follow McCarthy and Navigli (2007) in our choice of which grammatical relations to use as contextual features.

A variety of methods have been proposed for calculating distributional similarity. These have been shown to have differing characteristics (Lee 1999; Weeds et al., 2004) which make them useful for different applications or on different datasets. For the work reported here, we adopted the information-theoretic similarity measure due to Lin (1998), as this is in wide use and has been shown to perform well against other measures of similarity (Weeds and Weir, 2003; McCarthy and Navigli, 2009). Our thesaurus is represented as a collection of ‘nearest neighbour’ sets, one for each word in the thesaurus. For a given word, its set of nearest neighbours is ranked in order of decreasing similarity according to Lin’s distributional similarity measure. For a given number k, we may therefore talk about the top k neighbours of a target word, or its ‘k nearest neighbours’.

3. The Methods

3.1. The basic method

For the basic method, we obtained a number (k) of most similar words to the target word based on the thesaurus developed from the BNC using Lin’s measure of distributional similarity. The top k neighbours of a given target word thus provide the candidate substitutes for the Lexsub task. As a simple baseline, the topmost neighbour of each target word is selected as the ‘best’ substitute, regardless of context. For example, the topmost neighbour of the noun *film* in our thesaurus is *video*.

3.2. The overlap methods

These methods all use a simple overlap technique which assumes that a good synonym is substitutable in most contexts (Miller and Charles, 1991). We take the candidates supplied by the basic method as our starting point. As a proxy to substitutability, we re-rank the candidates (e.g. *video*, *movie*) using a measure of the overlap of the top k words related to the candidate word and those to the target word by one of a number of different statistical measures: distributional similarity, log-likelihood ratio (LLR) (Dunning 1993) and point-wise mutual information (PMI) (Church and Hanks 1990). LLR and PMI are statistical measures of association widely used for detecting collocations in computational linguistics. LLR has been recommended in preference to PMI because PMI over-estimates the level of association for rare words (Dunning 1993). As in the case of distributional similarity, both the LLR and PMI statistics were produced using word frequency data drawn from the written part of the British National Corpus. The overlap of the candidate's top k related words with those of the target word is calculated in the following ways:

3.2.1. Thesaurus overlap (Tol)

This method re-ranks candidates according to the overlap of the top k neighbours from the distributional thesaurus itself. For example, if the top 5 neighbours of *film* are as follows:

film: *video, movie, show, picture, series*

and the top 5 neighbours (according to the thesaurus) of each of the neighbours are as follows

video: film, tape, show, publication, photograph
movie: film, show, video, play, novel
show: exhibition, concert, festival, tour, display
picture: image, photograph, painting, portrait, story
series: show, programme, tour, event, set

then the overlap between *film* and *video* is 1 (they have *show* in common) whereas it is 2 between *film* and *movie* (*show* and *video* in common). So in this case, the method would re-rank *movie* above *video* as the most similar word to *film*. The overlap of the full top k candidates with the target would be calculated and the candidates then re-ranked using the overlap measure.

3.2.2. Log-Likelihood Ratio overlap (LLRol)

The candidates are re-ranked using the overlap of the top k words related to each candidate and to the target word by the log-likelihood ratio (LLR). Following the same example as above, we take the top 5 neighbours of *film* from the distributional thesaurus, but this time we compare the neighbours with the target in terms of the overlap of their top 5 ranked words[6] according to the LLR statistic with the top 5 words ranked by LLR for the target. The top five words ranked by LLR for the target *film* are as follows:

film: *cinema, television, movie, star, video*

and for each of the 5 nearest neighbours of *film*, the top 5 words ranked according to LLR are:

video: *recorder, tape, camera, film, audio*
movie: *film, star, star*(verb)*, cinema, horror, actor*
show: *exhibition, TV, gallery, artist, theatre*
picture: paint, frame, show, wall, flower
series: *NUMBER*[7]*, TV, publish*(verb)*, new*(adjective)*, television*

[6] For LLR and PMI we calculate over all words, regardless of the part of speech.

[7] The corpus was pre-processed to replace any numerical token with the token NUMBER.

In this case, both *movie* and *series* would be selected as synonyms of *film* as they both have one word (*cinema* and *television*, respectively) in common with the top 5 ranked words for *film*.

3.2.3. Point-Wise Mutual Information overlap (PMIol)

For this method, the candidates from the thesaurus are re-ranked using the overlap of the top k words related to the candidate and target word by point-wise mutual information (PMI). The PMIol method is thus exactly as for LLRol but using PMI as the statistic.

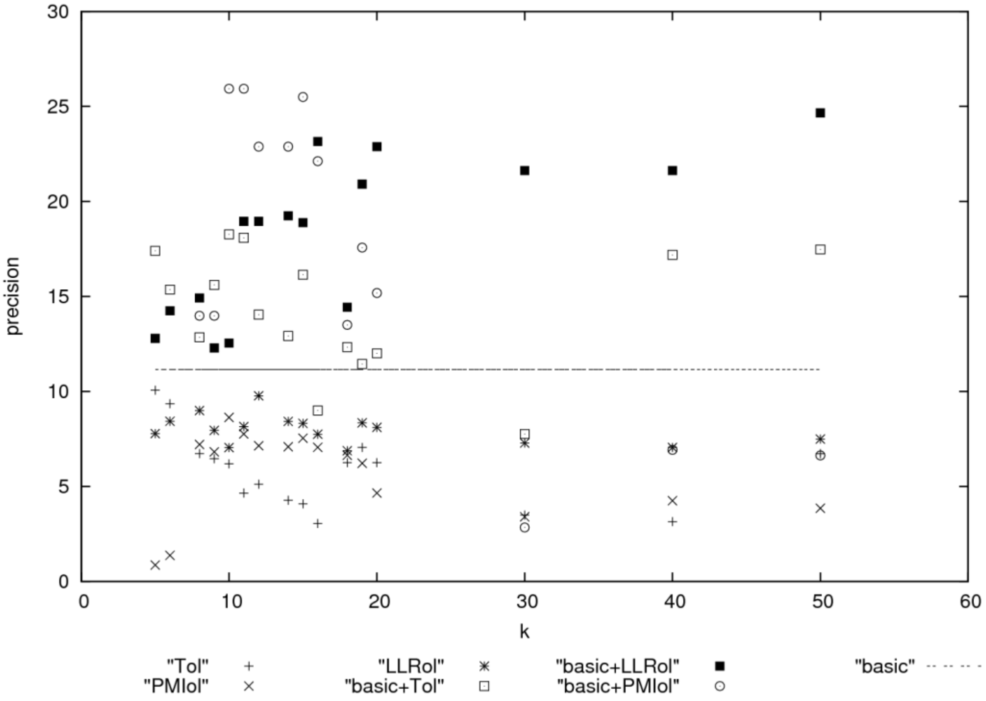

Figure 2. Precision of our methods on the lexsub development data.

3.3. Combining Methods

In practice, the overlap methods did not outperform the basic method (see figure 2 and the results below). That is, the distributional similarity measure provided by Lin is not improved by re-ranking nearest neighbours using any of the overlap methods. These methods however were used in conjunction with the basic method as follows.

We used the re-ranking methods described above as a veto on the decision to supply a substitute from the basic method. This is aimed at improving precision at the expense of coverage and therefore recall. We also tried a majority vote method, which allows each method (Tol, PMIol, LLRol and the basic method) one vote per item on the best substitute. The best substitute is then taken as that which has the majority vote from the four methods. In case of a split top score we select all substitutes having that score. We first used the development set to determine the best value of k for each combination method.

4. Results on the Lexsub data

We used the Lexsub development data to find an optimal value of k for each re-ranking method. For this, we compared the precision on the development data using k=5..20, 30, 40 and 50. In figure 2 we summarise the results obtained when varying the number of neighbours (k) for re-ranking. The baseline (basic) method is represented by the horizontal line, since we always pick the first neighbour regardless of the value of k.

The Tol, PMIol and LLRol methods do not improve precision over the basic method and are therefore represented by points beneath the baseline. In contrast, the basic+Tol, basic+LLol and basic+PMIol combination methods, which restrict attempts to those items where the respective overlap methods (Tol, PMIol and LLRol) are in agreement with the basic method tend to increase precision. For basic+Tol and basic+PMIol, the optimal value of k seems to be at 10. For basic+LLRol there is an outlier at k = 50 but other than that the data points indicate k = 16 as a good setting. The exact results for the optimal value of k are shown in table 1.

From the results we see that the combination methods increase the precision, albeit at the expense of recall. Since we do not use the local context in our experiments, we report an upper bound on the data which shows the optimal performance that could be obtained using an oracle to pick the best of the candidates from the distributional thesaurus. The best candidate is the one which has maximum frequency from the annotations over the ten test sentences for a given word and part of speech. Thus, the optimum that our methods could achieve without taking context into account is 27.39. Although the basic method covers more items, the combinations are able to improve precision considerably on the development data. The majority vote technique outperforms the basic method on the development data in terms of precision and recall but the difference is not statistically significant. The other combined methods are significantly better in terms of precision. The optimal setting for k for each method was then used to evaluate the method on the test data, with results as shown in table 2.

Method (optimal k)	Precision	Recall	Number attempted
Basic Method	11.15	11.12	294
Basic + Tol (10)	18.27	4.95	80
Basic + LLRol (16)	23.15	6.59	84
Basic + PMIol (10)	25.93	5.27	60
Majority Vote (6)	11.77	11.73	294
Upper bound	27.39	27.39	295

Table 1. lexsub best measure on development data

Method (optimal k)	Precision	Recall	Number attempted
Basic Method	8.82	8.51	1636
Basic + Tol (10)	11.47	5.17	765
Basic + LLRol (16)	13.99	3.16	383
Basic + PMIol (10)	10.73	2.02	328
Majority Vote (6)	8.15	7.86	1636
Upper bound	27.31	27.31	1696

Table 2. lexsub best measure on test data

There are 295 items in the development data, although we only answer a small selection (60-84 items) with these methods. The precision is very high and in some cases near the upper bound, which is very encouraging given that we find the candidates automatically, without manually produced resources. The increase in precision over the test data is also promising. We note that the optimal settings for combination and k on the development data are not the same as for the test data. That is, the optimal combination and value of k differs, so that the results would be improved with a larger development set[8].

The best precision reached by a system participating in the Lexsub task was 12.90. Our best combination achieved a precision of 13.99, although our method has lower coverage and therefore recall. The best participating system used Roget's thesaurus to generate candidate substitutes and information about local context to select a best substitute. In contrast, our method uses only corpus data and does not take local context into account. We plan to use context with corpus-based, distributional methods in the future. The WordNet baseline provided by the task organisers, which used synonyms and sense frequency information from WordNet 2.1, had precision and recall at 9.95. Our automatic method has improved precision compared to this manually produced thesaurus. From manual inspection, many of the automatically generated synonyms look reasonable. For example, for LLRol we have *crazy* for *mad*, *apparently* for *seemingly*, *rhythm* for *beat*, *hit* for *strike*. However, there are also cases of co-hyponyms such as *go* for *come* and *red* for *blue* which have not been filtered by the combination methods.

5. Related Work

While there are promising approaches to synonym detection which use hand crafted knowledge or training data, our approach makes use of automatically acquired lists of candidate synonyms using unsupervised distributional similarity methods. There are other promising corpus based methods. Landauer and Dumais (1997) applied Latent Semantic Analysis to the problem of classifying synonyms of a target word. They used 80 standard TOEFL multiple-choice test questions as their test data attaining good accuracy. The best results on that dataset have been achieved by Turney et al (2003). Turney’s approach combines the probability distributions generated by four independent modules using a novel product rule which makes use of manually produced resources and corpus based methods.

We note that the Lexsub task is not posed as a multiple-choice test. Finding a best lexical substitute for a given problem word is thus more challenging than that of choosing a correct synonym given a choice of four targets, since it is necessary to generate and rank possible candidates. Systems that can perform the Lexsub task are relevant to lexicography because they generate the synonyms which could be useful for thesaurus construction.

6. Conclusion

We have presented a method for determining when the highest ranked distributional neighbour of a given target work is likely to be a good synonym for that word. It has been demonstrated that this approach can improve the precision of finding synonyms, albeit at the expense of recall. We report the best precision results on the Lexsub dataset, which is particularly encouraging as our method does not require manually produced thesauruses and does not yet exploit the context. We acknowledge that we have yet to try our method on other synonym tasks and we hope to do so in the near future.

The work described in this paper does not address the problem of selecting a best substitute on the basis of the context in which the target word appears, yet it should be possible to improve performance by making use of contextual information to rank the set of available candidates. Most systems participating in the Lexsub task made use of

[8] We do not report the optimal value of k on the test data as parameters should be determined on held out ‘development’ data.

n-gram language models for this purpose (McCarthy and Navigli 2009). We are currently exploring an approach that uses a second distributional thesaurus constructed from contextual features representing word co-occurrence within a sentence. In the resulting thesaurus, words with high similarity are those that co-occur with very similar sets of words and these scores can therefore be used to measure the degree of match between candidate synonyms and the target context.

References

Briscoe, E. and Carroll, J. (2002). ‘Robust accurate statistical annotation of general text’. In *Proceedings of LREC-2002*, 1499–1504, Las Palmas, Canary Islands, Spain.

Church, K. and P. Hanks. (1990). ‘Word association norms, mutual information and lexicography’. *Computational Linguistics*, 19 (2). 263–312.

Curran, J. and M. Moens. (2002). ‘Improvements in automatic thesaurus extraction’. In *Proceedings of the ACL-SIGLEX Workshop on Unsupervised Lexical Acquisition*. 59-66. Philadelphia.

Dunning, T. (1993). ‘Accurate methods for the statistics of surprise and coincidence’. In *Computational Linguistics,* 19 (1). 61–74.

Erk, K., D. McCarthy and N. Gaylord. (2009) ‘Investigations on Word Senses and Word Usages’. In *Proceedings of ACL-IJCNLP*. 10–18, Singapore.

Fellbaum, C. (ed.). (1998) *WordNet: An Electronic Lexical Database*. MIT Press.

Grefenstette, G. (1994). ‘Corpus-derived first-, second- and third-order word affinities’. In *Proceedings of Euralex*. 279–280. Amsterdam.

Harris, Z. (1968). *Mathematical Structures of Language*. Wiley: New York.

Kilgarriff, A., P. Rychly, P. Smrz and D. Tugwell. (2004). ‘The Sketch Engine’. In *Proceedings Euralex.* Lorient, France, July. 105–116.

Landauer, T.K. and S. T Dumais. (1997). ‘A solution to Plato’s problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge’. In *Psychological Review*, 104 (2). 211–240.

Lee, L. (1999). ‘Measures of distributional similarity’. In *Proceedings of the 37th Annual Meeting of the ACL*. 23–32, College Park, Maryland.

Leech, G. (1992). ‘100 million words of English: the British National Corpus’. In *Language Research*, 28 (1). 1–13.

Lin, D. (1998). ‘Automatic retrieval and clustering of similar words’. In *Proceedings of COLING-ACL 98,* 768–774, Montreal, Canada.

McCarthy, D. and R. Navigli. (2007). ‘SemEval-2007 task 10: English lexical substitution task’. In *Proceedings of SemEval-2007*. 48–53, Prague.

McCarthy, D. and R. Navigli. (2009) ‘The English lexical substitution task’. In *Language Resources and Evaluation Special Issue on Computational Semantic Analysis of Language*, 43 (3). 139–159.

Miller, G. A. and W. G. Charles. (1991) ‘Contextual correlates of semantic similarity’. In *Language and Cognitive Processes,* 6 (1). 1–28.

Navigli, R. (2009). ‘Word Sense Disambiguation: a Survey’. In *ACM Computing Surveys*, 41 (2). 1–69, ACM Press.

Palmer, M., O. Babko-Malaya and H.T. Dang. (2004). ‘Different sense granularities for different applications’. In *Proceedings of the 2nd Workshop on Scalable Natural Language Understanding Systems in HLT/NAACL*, 49–56, Boston, MA.

Palmer, M., H.T. Ng and H. T. Dang. (2006). ‘Evaluation of WSD systems’. In *Word Sense Disambiguation: Algorithms and Applications*, E. Agirre and P. Edmonds (Eds.), 75-106.

Rychlý, P. and A. Kilgarriff. (2007). ‘An efficient algorithm for building a distributional thesaurus (and other Sketch Engine developments)’. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, 41-44. Prague.

Sharoff, S. (2006). ‘Open-source corpora: Using the net to fish for linguistic data’. *International Journal of Corpus Linguistics*, 11 (4), 435–462.

Turney, P. D.; M. L. Littman; J. Bigham; and V. Shnayder. (2003). ‘Combining independent modules to solve multiple-choice synonym and analogy problems’. In *Proceedings of RANLP-03*, 482–489.

Weeds, J. (2003). *Measures and applications of lexical distributional similarity*. PhD thesis, Department of Informatics, University of Sussex.

Weeds, J. and D. Weir. (2003). ‘Finding and Evaluating Sets of Nearest Neighbours’. In *Proceedings of the 2nd Conference of Corpus Linguistics*, Lancaster.

Weeds, J.; D. Weir and D. McCarthy. (2004). ‘Characterising measures of lexical distributional similarity’. In *Proceedings of the 20th International Conference for Computational Linguistics*, 10-15, Geneva.

Yuret, D. Ku. (2007). ‘Word sense disambiguation by substitution’. In Proceedings *of SemEval-2007*, 207–214, Prague.