Using Centrality Algorithms on Directed Graphs for Synonym Expansion

Ravi Sinha and **Rada Mihalcea**
Computer Science and Engineering
University of North Texas
ravisinha@my.unt.edu, rada@cs.unt.edu

Abstract

This paper presents our explorations in using graph centrality measures to solve the synonym expansion problem. In particular, we use the concept of directional similarity to derive directed graphs on which we apply centrality algorithms to identify the most likely synonyms for a target word in a given context. We show that our method can lead to performance comparable to the state-of-the-art.

Introduction

Synonym expansion can be viewed as a specific type of word sense disambiguation in that it attempts to find the correct meaning of a word by identifying its synonyms (or substitutes) in a given context. Unlike word sense disambiguation, which typically relies on predefined sense inventories, synonym expansion (or lexical substitution) is more flexible as it can define the meaning of a word "on the fly," based on its current context.

Given a sentence, for example *He was a **bright** boy*, the task is to find synonyms that could replace the word *bright* without changing the meaning of the sentence. Several methods to solve this problem have already been proposed, see for instance (McCarthy & Navigli 2007) for an overview of several systems that participated in the SEMEVAL lexical substitution task, or (Sinha & Mihalcea 2009) for a comparative exploration of different resources and tools.

The approach proposed in this paper relies on a combination of graph centrality measures and *directional similarity*, to identify the most likely synonyms for a target word in a given context. Through experiments, we show that this unsupervised method is competitive with some of the best results obtained so far on this task.

The paper is organized as follows. First, we describe the task of synonym expansion in more detail, along with defining the evaluation metrics as well as the data sets that have been used for this task. We then discuss the basics of directional similarity and graph centrality, focusing on degree, PAGERANK and biased-PAGERANK. Next, we describe the experiments and evaluations, and finally conclude with an analysis of the results and possibilities for future work.

Synonym Expansion in Context

Contextual synonym expansion, also known as lexical substitution (McCarthy & Navigli 2007), is the task of replacing a certain word in a given context with another, suitable word. See for example the four sentences from Table 1, drawn from the development data from the SEMEVAL-2007 lexical substitution task. In the first sentence, assuming we choose *bright* as the target word, a suitable substitute could be *brilliant*, which would both maintain the meaning of the target word and at the same time fit the context.

Sentence	Target	Synonym
The sun was **bright**.	bright	brilliant
He was **bright** and independent.	bright	intelligent
His feature **film** debut won awards.	film	movie
The market is **tight** right now.	tight	pressured

Table 1: Examples of synonym expansion in context

The task arose from the idea of trying to test word sense disambiguation systems without a predetermined sense inventory, since there is no clear consensus as to which particular sense inventory is appropriate for a given task, and how coarse-grained or how fine-grained such an inventory should be for an automatic system to be useful in practice.

Data

The data used for the evaluation of systems participating in the SEMEVAL-2007 lexical substitution task consisted of 2010 examples for 201 words covering all open class parts-of-speech (i.e., nouns, verbs, adjectives and adverbs), keeping in view a preference for polysemous words. The examples were extracted from the English Internet Corpus (Sharoff 2006), and human annotations were collected from five annotators. In our experiments, we use the same trial and test datasets as in the original task evaluations.

Evaluation Metrics

We use the same evaluation metrics as used for the lexical substitution task. Specifically, we adopt the BEST and OUT-OF-TEN (OOT) precision and recall scores from (McCarthy & Navigli 2007). We allow as many substitutes as the algorithm feels fit for the context, and the credit is given depending on the number of annotators that picked that substitute as well as the number of annotator responses for the item, and the number of answers provided by the system.

the BEST scorer gives credit to only one best answer. If the system provides several answers, the credit is divided among them. Formally, if i is an item in the set of instances I, and T_i is the multiset of gold standard synonym expansions from the human annotators for i, and a system provides a set of answers S_i for i, then the BEST score for item i is:

$$best\ score(i) = \frac{\sum_{s \in S_i} frequency(s \in T_i)}{|S_i| \cdot |T_i|} \quad (1)$$

Precision is calculated by summing the scores for each item and dividing by the number of items that the system attempted whereas recall divides the sum of scores for each item by $|I|$. Thus:

$$best\ precision = \frac{\sum_i best\ score(i)}{|i \in I : defined(S_i)|} \quad (2)$$

$$best\ recall = \frac{\sum_i best\ score(i)}{|I|} \quad (3)$$

The OOT scorer allows up to ten system responses and does not divide the credit for an answer by the number of system responses.

$$oot\ score(i) = \frac{\sum_{s \in S_i} frequency(s \in T_i)}{|T_i|} \quad (4)$$

$$oot\ precision = \frac{\sum_i oot\ score(i)}{|i \in I : defined(S_i)|} \quad (5)$$

$$oot\ recall = \frac{\sum_i oot\ score(i)}{|I|} \quad (6)$$

For both the BEST and OOT measures, in addition to the regular (*normal*) score, we also report a *mode* score, which is computed taking into account only the most frequent response among the annotators (no *mode* is calculated for those items that do not have a most frequent answer).

The DSIM Algorithm

Our algorithm (DSIM, for Directional Similarity) consists of several steps, which combine centrality algorithms on directed graphs and measures of directional similarity.

Given a sentence and a target word, we start by collecting all the synonyms for the target word, as well as collecting all the context words. The synonyms are generated using several different resources, including WordNet, Encarta, Roget Thesaurus, TransGraph and distributional similarity; see (Sinha & Mihalcea 2009) for details on these resources. From these, we work with those individual and combined resources that were found to work best in previous work (Sinha & Mihalcea 2009): Encarta, WordNet, a combination of Encarta and WordNet picking candidates present in both resources, a combination picking candidates present in either one of the resources, candidates present in two or more out of all the resources, and finally candidates present in three or more resources.

The entire set of candidate synonyms, along with all the open-class words in the surrounding context, are used to generate the vertices in the graph. To draw edges between words, we use a measure of directional similarity, as described below. The edges are directed, with the orientation of the edge being determined by the direction of the similarity of the words in the pair. The edge weight is the actual value of the similarity. Figure 1 shows an example of the graph generated for a sample sentence.

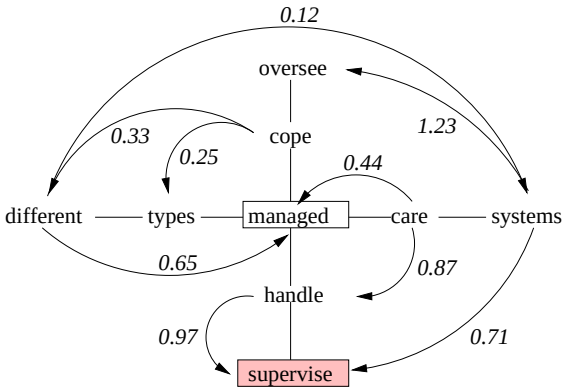

Figure 1: A sample sentence and the associated directed graph, for the sentence *There are different types of managed care systems*, with the target word being **manage**

Directional Similarity

Directional similarity, as opposed to the traditional, symmetric similarity, is a new concept introduced and discussed in (Michelbacher, Evert, & Schütze 2007; Leong, Mihalcea, & Hassan 2010).

The concept of *salience* has been long discussed, for example in (Durkin & Manning 1989). The traditional school of thought has always maintained that if two words are related to each other (regardless of whether we talk about relatedness or similarity), then that relationship is symmetric, and any method of quantifying their relatedness or similarity as a concrete number assigns the same quantity to the relationship from the first word to the second word as to the relationship from the second word to the first word.

There is however a new school of thought that pro-

motes the concept of *directional similarity*, and tries to incorporate the salience of words in intra-word relationships. To illustrate, consider the word *Clinton*, which makes us automatically think of *president*, but the reverse is not true: the word *president* more often than not does *not* make us think of *Clinton*. Thus, assuming a hypothetical metric $DSim$ that accounts for the directional similarity between words, then $DSim(Clinton, president) > DSim(president, Clinton)$. In other words, *Clinton* is more related or similar to *president* than the other way around. We can rephrase it by saying that *Clinton* is more salient than *president* in the relatedness or similarity relationship between the two words.

Formally, given two words $w1$ and $w2$, we define:

$$DSim(w1, w2) = \frac{C_{12}}{C_1} Sim(w1, w2) \quad (7)$$

where

$$Sim(w1, w2) = Cos.Sim(ESA(w1), ESA(w2)) \quad (8)$$

C_{12} is the number of articles in the British National Corpus that contain both words $w1$ and $w2$, and C_1 is the number of articles that contain $w1$. In our implementation, $Sim(w1, w2)$ is the cosine similarity between the Explicit Semantic Analysis (ESA) vectors of the two words[1] (Gabrilovich & Markovitch 2007). Note that other similarity or relatedness metrics can also be used, such as Latent Semantic Analysis (LSA) (Deerwester *et al.* 1990) or others.

Since the direction of similarity is not known apriori, for each pair of words we calculate two $DSim$ values, corresponding to the two possible directions that can be established between the words. The second value can be determined by using in the denominator of equation 7 the number of articles in the corpus that contain the second word. Out of these two values, the higher value determines the direction of relatedness, with the direction set from the more salient word in the relationship to the less salient word. Formally, if $\text{DSIM}(w1, w2) > \text{DSIM}(w2, w1)$, we say the direction is from $w1$ to $w2$.[2]

Graph Centrality Algorithms

Given the graph representation of an input sentence, including the context words as well as the candidate synonyms for the target word, we use graph-centrality algorithms to determine the relative importance of the nodes in the graph, and thus find the synonyms that are most likely to fit the given context.

[1]ESA is a novel approach for computing semantic relatedness between words. In contrast to using any human-generated hierarchies or data to compute this value, ESA attempts to represent meanings of words or texts in a high-dimensional vector space of concepts derived from Wikipedia.

[2]When $\text{DSIM}(w1, w2) = \text{DSIM}(w2, w1)$, we only use one direction $w2$ to $w1$.

Resource	best normal	best mode	oot normal	oot mode
UNDIR(LSA), DEG				
encarta	0.7	0.8	22.7	29.0
wordnet	3.2	3.2	17.9	24.0
e and w	5.6	5.4	15.6	18.7
e or w	3.4	4.0	31.1	34.5
any2	3.1	3.5	31.5	35.7
any3	6.2	7.0	31.2	42.7
UNDIR(LSA), PR				
encarta	1.2	1.5	23.9	29.6
wordnet	3.4	3.9	18.9	24.8
e and w	6.0	5.5	20.1	22.2
e or w	3.4	4.2	30.1	33.8
any2	3.6	4.0	31.5	35.7
any3	6.5	7.2	31.2	42.7
UNDIR(ESA), DEG				
encarta	6.8	7.8	32.0	42.5
wordnet	6.8	8.7	21.7	27.7
e and w	9.4	10.2	20.7	26.7
e or w	5.1	6.8	30.7	37.9
any2	3.5	3.9	28.9	38.3
any3	7.4	10.7	36.7	49.5
UNDIR(ESA), PR				
encarta	7.1	8.7	32.0	41.3
wordnet	6.9	8.7	21.7	27.7
e and w	**9.6**	**11.2**	19.8	26.7
e or w	5.3	7.8	30.8	38.3
any2	3.9	5.3	29.3	39.8
any3	7.3	10.7	36.8	50.5
DIR(ESA), DEG				
encarta	4.5	4.4	22.3	27.7
wordnet	5.8	5.3	18.6	24.4
e and w	5.1	4.9	20.7	26.7
e or w	4.8	3.9	21.2	27.2
any2	4.6	4.4	22.4	29.1
any3	6.2	9.2	28.8	40.8
DIR(ESA), PR				
encarta	4.4	4.4	31.0	37.9
wordnet	6.0	5.3	21.9	29.1
e and w	7.8	6.3	19.8	26.7
e or w	3.9	3.9	29.0	35.4
any2	4.0	3.9	28.1	37.4
any3	5.8	7.8	**38.5**	**53.9**
DIR(ESA), BPR				
encarta	4.6	4.4	30.6	36.9
wordnet	6.4	6.3	22.0	29.1
e and w	7.5	5.8	19.8	26.7
e or w	3.6	2.9	29.9	36.4
any2	4.3	5.3	27.2	35.4
any3	5.8	7.8	37.9	53.4

Table 2: Experiments on development data for LSA and ESA; directed (DIR) and undirected (UNDIR) graphs; degree (DEG), PAGERANK (PR) and biased PAGERANK (BPR).

The basic idea implemented by a graph centrality algorithm is that the "importance" of a node in a graph can be determined by taking into account the relation of the node with other nodes in the graph. In our experiments, we use two centrality algorithms: degree and PAGERANK (Brin & Page 1998).

For directed graphs, we define the degree of a node as the difference between the sum of the weights of all the incoming edges to that node (indegree) and the sum of the weights of all the outgoing edges from that node (outdegree). The intuition behind this is that if a lot of vertices point to a certain vertex in the graph, then it must be important.

For weighted graphs, we calculate the degree by taking into account the weights on the edges:

$$Degree(V_a) = \sum_{(V_b,V_a)\in E} w_{ba} - \sum_{(V_a,V_b)\in E} w_{ab} \quad (9)$$

where $G = (V, E)$ is a graph with vertices $v \in V$ and directed edges $e \in E$, and w_{ab} is the weight on the edge between V_a and V_b.

The other graph centrality algorithm we consider is PAGERANK. The main idea implemented by PAGERANK is that of "voting" or "recommendation." When one vertex links to another one, it is basically casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex. Moreover, the importance of the vertex casting a vote determines how important the vote itself is, and this information is also taken into account by the ranking algorithm. The PAGERANK score associated with a vertex V_a is defined using a recursive function:

$$PageRank(V_a) = (1-d) + d * \sum_{(V_b,V_a)\in E} \frac{PageRank(V_b)}{|Outdegree(V_b)|} \quad (10)$$

where d is a parameter that is set between 0 and 1. The typical value for d is 0.85 (Brin & Page 1998), and this is the value we are using in our implementation.

In a weighted graph, the decision on what edge to follow during a random walk is also taking into account the weights of outgoing edges, with a higher likelihood of following an edge that has a larger weight (Mihalcea & Tarau 2004). Given a set of weights w_{ab} associated with edges connecting vertices V_a and V_b, the weighted PAGERANK score is determined as:

$$PageRank(V_a) = (1-d) + d \sum_{(V_b,V_a)\in E} \frac{w_{ba} PageRank(V_b)}{\sum_{(V_c,V_b)\in E} w_{cb}} \quad (11)$$

PAGERANK in its traditional sense corresponds to a uniform probability distribution among the vertices in the graph. Instead, biased PAGERANK, first mentioned in (Brin *et al.* 1998) and (Haveliwala 1999) and further referenced in (Haveliwala 2003), takes this idea further by introducing the concept of relative importance of the vertices. Instead of assigning the same probability to each vertex that a random surfer could potentially jump to, biased PAGERANK allows a certain "bias" toward certain vertices. This is done by multiplying the corresponding contributing score of a vertex by its bias weight, determined by whether that vertex belongs to a word in context or whether it is a synonym.

Resource	best normal	best mode	oot normal	oot mode
encarta	8.3	12.4	32.9	41.8
wordnet	9.1	13.6	21.8	27.2
e and w	10.1	14.1	20.3	25.5
e or w	8.6	14.1	36.2	45.8
any2	7.1	11.4	33.2	42.3
any3	9.3	14.1	30.9	44.0

Table 3: Experiments on development data, as reported in previous work (Sinha & Mihalcea 2009); the results were obtained mostly with a statistical method using Google Web 1T

Experiments and Evaluation

We started our evaluations by running several experiments on a development data set, to determine the resources and methods that provide the best results.

Table 2 shows the results obtained on the development data set using (1) each of the six resources described before: *WordNet*, *Encarta*, WordNet and Encarta (*w and e*), WordNet or Encarta (*w or e*), candidates present in two or more out of all the resources, including also Roget, Transgraph and the distributional similarity (*any2*), and finally candidates present in three or more resources (*any3*); (2) LSA or ESA; (3) directed or undirected graphs; (4) PAGERANK or degree; (5) unbiased or biased graph centrality, with a bias set toward the words in the context. Moreover, in Table 3, we also compare our results with the previous work done and presented in (Sinha & Mihalcea 2009).

Several comparative analyses can be made with these tables. Looking at Table 2, we start by comparing ESA and LSA. It can be seen that in general, on average ESA tends to perform better than LSA. This conclusion can be drawn based on the results for undirected PAGERANK and degree, looking at results obtained between the ESA and LSA variants.

Our next comparison is made between directed graphs and undirected graphs, i.e. directional similarity emphasizing salience and the traditional, symmetric similarity. When used in conjunction with the PAGERANK algorithm, and applied on a large number of candidate synonyms, the directional similarity outperforms the symmetric measure by a significant margin, as seen in the OOT scores for *any3* between the tables for DIR(ESA), PR and DIR(ESA), BPR in contrast to UNDIR(ESA), PR.

We next focus on whether it is worthwhile to run PAGERANK or a simple degree computation suffices.

Looking at Table 2, it seems that for directed graphs PAGERANK performs better than degree, especially for the OOT measure. For undirected graphs, however, the differences are not too pronounced. Moreover, evaluations of the biased PAGERANK show that the performance is comparable with the simple PAGERANK.

From these experiments, we can conclude that for selecting a large number of synonyms (OOT), the best setting consists of using a directional similarity calculated using ESA, combined with PAGERANK run on the resulting directed graph to select the most appropriate synonyms. When only one synonym is to be selected (BEST), the use of PAGERANK on an undirected weighted graph using an ESA similarity gives the best results.

As an additional experiment, we also tried to reverse the directions of the edges, i.e., made them point from the less salient word to the more salient word. The results were markedly lower than those for the original directionality, which proves that the use of directed edges is effective, and the edges should indeed point from the more salient word to the less salient word in a word pair.

Finally, comparing our results with those reported in (Sinha & Mihalcea 2009) on the same development data set, as shown in Table 3, we can conclude that a method based on Google Web 1T performs very well for selecting the top (BEST) candidate, while our graph-based method performs better for selecting the top ten (OOT) candidates. As one example, using the resource *any3*, PAGERANK on directed graphs surpasses the top result in (Sinha & Mihalcea 2009) by an absolute 2% in the OOT-NORMAL metric and 8% in the OOT-MODE metric.

Evaluations on Test Data

Using the settings determined earlier on the development data set, we also run experiments on the test data, with results shown in Table 4. For comparison, we also show in Table 5 the results for the unsupervised methods reported in (Sinha & Mihalcea 2009). The results reported by our system are better than the previous results for the OOT metric.

Finally, in Table 6, we show the results obtained by various teams participating in the original lexical substitution task as reported in (McCarthy & Navigli 2007).

Several of these systems used a combination of expensive machine learning methods to solve the problem, as opposed to our relatively simple and straightforward approach of graph centrality. Most of the systems used only one lexical resource, and a few used two resources. Google Web 1T was the most common resource to gather counts for contextual fitness. KU as described in (Yuret 2007) used a statistical language model based on the Google Web 1T five-grams dataset to compute probabilities for all the synonyms and worked with the Roget thesaurus. UNT (Hassan *et al.* 2007) used WordNet and Encarta, along with back-and-forth translations collected from commercial translation engines, and N-gram-based models calculated on the Google Web 1T corpus. IRST2 (Giuliano, Gliozzo, & Strapparava 2007) used synonyms from WordNet and the Oxford American Writer Thesaurus, and ranked them based on the Google 1T five-grams corpus. HIT (Zhao *et al.* 2007), used WordNet to extract the synonyms and used Google queries to collect the counts, only looking at words close to the target in context. Another high-scoring system was MELB (Kim & Baldwin 2007), which used WordNet, Google queries, and combined the two with a heuristic taking into account the length of the query and the distance between the target word and the synonym inside the lexical resource.

Resource	best normal	best mode	oot normal	oot mode
encarta	5.4	8.3	38.2	44.9
wordnet	8.1	12.1	30.1	40.1
e and w	11.2	9.8	27.5	38.0
e or w	5.4	7.6	36.3	45.5
any2	5.9	9.7	35.1	47.4
any3	7.7	15.4	50.7	66.3

Table 4: Results obtained by our graph method on the test data.

Metric	Individual resource	F1	Combined resource	F1
$best, normal$	wordnet	10.1	e and w	12.8
$best, mode$	wordnet	16.0	any3	19.7
$oot, normal$	encarta	43.2	e or w	43.7
$oot, mode$	encarta	55.3	e or w	58.4

Table 5: Results on the test data for the unsupervised method reported in (Sinha & Mihalcea 2009)

As can be determined from these tables, our system performs on par with the best systems, while taking a completely different approach, which exploits graphs that encode relations between words in the text, instead of very large resources such as Google Web 1T.

Conclusion and Future Work

In this paper, we presented our explorations in using graph-based algorithms and directional similarity in an attempt to solve the problem of automatic synonym expansion. Through several experiments, we showed the utility and potential of centrality algorithms applied on directed graphs modeling salience in intra-word relationships, and showed that this unsupervised graph-based method can lead to results competitive with the state-of-the-art.

As a future point of interest, we would like to employ other measures of directional similarity, which might prove to be less resource intensive than the one utilized in this work, and might improve the results. Two such potential options are presented in (Michelbacher, Evert, & Schütze 2007) and (Martin & Azmi-Murad 2005).

Resource	best normal	best mode	oot normal	oot mode
Systems				
IRST2	6.95	20.33	68.96	58.54
UNT	12.77	20.73	49.19	66.26
KU	12.90	20.65	46.15	61.30
IRST1	8.06	13.09	41.21	55.28
MELB	13.35	14.00	-	-
USYD	11.05	17.93	35.51	42.96
SWAG2	-	-	36.16	48.01
HIT	11.35	18.86	33.88	46.91
SWAG1	-	-	34.13	45.54
TOR	2.98	2.98	11.19	14.63
Baselines				
WordNet	9.95	15.28	29.52	40.57
Lin	8.68	14.45	27.20	39.82
L1	7.96	13.14	23.65	35.52
Lee	6.86	11.15	19.72	29.32
Jaccard	6.71	10.99	17.90	26.44
Cos	4.98	7.52	13.82	20.48

Table 6: Results obtained by the teams participating in the lexical substitution task SEMEVAL2007

Acknowledgments

This material is based in part upon work supported by the National Science Foundation CAREER award #0747340. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Brin, S., and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.* 30(1-7):107–117.

Brin, S.; Motwani, R.; Page, L.; and Winograd, T. 1998. What can you do with a web in your pocket? *IEEE Data Engineering Bulletin* 21(2):37–47.

Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41:391–407.

Durkin, K., and Manning, J. 1989. Polysemy and the subjective lexicon: Semantic relatedness and the salience of intraword senses. *Journal of Psycholinguistic Research* 18:577–612.

Gabrilovich, E., and Markovitch, S. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of The Twentieth International Joint Conference for Artificial Intelligence*, 1606–1611.

Giuliano, C.; Gliozzo, A.; and Strapparava, C. 2007. Fbk-irst: Lexical substitution task exploiting domain and syntagmatic coherence. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, 145–148. Prague, Czech Republic: Association for Computational Linguistics.

Hassan, S.; Csomai, A.; Banea, C.; Sinha, R.; and Mihalcea, R. 2007. Unt: Subfinder: Combining knowledge sources for automatic lexical substitution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, 410–413. Prague, Czech Republic: Association for Computational Linguistics.

Haveliwala, T. 1999. Efficient computation of pagerank. Technical report.

Haveliwala, T. H. 2003. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering* 15:784–796.

Kim, S. N., and Baldwin, T. 2007. Melb-kb: Nominal classification as noun compound interpretation. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, 231–236. Prague, Czech Republic: Association for Computational Linguistics.

Leong, B.; Mihalcea, R.; and Hassan, S. 2010. Text mining for automatic image tagging. In *In Proceedings of the International Conference on Computational Linguistics (COLING 2010)*.

Martin, T. P., and Azmi-Murad, M. 2005. An incremental algorithm to find asymmetric word similarities for fuzzy text mining. In *WSTST*, 838–847.

McCarthy, D., and Navigli, R. 2007. Semeval-2007 task 10: English lexical substitution task. In *In Proceedings of the 4th workshop on Semantic Evaluations (SemEval-2007)*, 48–53.

Michelbacher, L.; Evert, S.; and Schütze, H. 2007. Asymmetric association measures. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP 2007*.

Mihalcea, R., and Tarau, P. 2004. TextRank – bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*.

Sharoff, S. 2006. Creating general-purpose corpora using automated search engine queries. In *WaCky! Working papers on the Web as Corpus. Gedit*.

Sinha, R., and Mihalcea, R. 2009. Combining lexical resources for contextual synonym expansion. In *Proceedings of the International Conference RANLP-2009*, 404–410. Borovets, Bulgaria: Association for Computational Linguistics.

Yuret, D. 2007. Ku: Word sense disambiguation by substitution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, 207–214. Prague, Czech Republic: Association for Computational Linguistics.

Zhao, S.; Zhao, L.; Zhang, Y.; Liu, T.; and Li, S. 2007. Hit: Web based scoring method for english lexical substitution. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, 173–176. Prague, Czech Republic: Association for Computational Linguistics.