A Semantic Network Approach to Measuring Relatedness

Brian Harrington
Oxford University Computing Laboratory
`brian.harrington@comlab.ox.ac.uk`

Abstract

Humans are very good at judging the strength of relationships between two terms, a task which, if it can be automated, would be useful in a range of applications. Systems attempting to solve this problem automatically have traditionally either used relative positioning in lexical resources such as WordNet, or distributional relationships in large corpora. This paper proposes a new approach, whereby relationships are derived from natural language text by using existing NLP tools, then integrated into a large scale semantic network. Spreading activation is then used on this network in order to judge the strengths of all relationships connecting the terms. In comparisons with human measurements, this approach was able to obtain results on par with the best purpose built systems, using only a relatively small corpus extracted from the web. This is particularly impressive, as the network creation system is a general tool for information collection and integration, and is not specifically designed for tasks of this type.

1   Introduction

The ability to determine semantic relatedness between terms is useful for a variety of NLP applications, including word sense disambiguation, information extraction and retrieval, and text summarisation (Budanitsky and Hirst, 2006). However, there is an important distinction to be made between semantic *relatedness* and semantic *similarity*. As (Resnik, 1999) notes, "Semantic similarity represents a special case of semantic relatedness: for example, cars and gasoline would seem to be more closely related than, say, cars and bicycles, but the latter pair are certainly more similar". (Budanitsky and Hirst, 2006) further note that "Computational applications typically require relatedness rather than just similarity; for example, money and river are cues to the in-context meaning of bank that are just as good as trust company".

Systems for automatically determining the degree of semantic relatedness between two terms have traditionally either used a measurement based on the distance between the terms within WordNet (Banerjee and Pedersen, 2003; Hughes and Ramage, 2007), or used co-occurrence statistics from a large corpus (Mohammad and Hirst, 2006; Padó and Lapata, 2007). Recent systems have, however, shown improved results using extremely large corpora (Agirre et al., 2009), and existing large-scale resources such as Wikipedia (Strube and Ponzetto, 2006).

In this paper, we propose a new approach to determining semantic relatedness, in which a semantic network is automatically created from a relatively small corpus using existing NLP tools and a network creation system called ASKNet (Harrington and Clark, 2007), and then spreading activation is used to determine the strength of the connections within that network. This process is more analogous to the way the task is performed by humans. Information is collected from fragments and assimilated into a large semantic knowledge structure which is not purposely built for a single task, but is constructed as a general resource containing a wide variety of information. Relationships represented within this

structure can then be used to determine the total strength of the relations between any two terms.

2 Existing Approaches

2.1 Resource Based Methods

A popular method for automatically judging semantic distance between terms is through WordNet (Fellbaum, 1998), using the lengths of paths between words in the taxonomy as a measure of distance. While WordNet-based approaches have obtained promising results for measuring semantic similarity (Jiang and Conrath, 1997; Banerjee and Pedersen, 2003), the results for the more general notion of semantic relatedness have been less promising (Hughes and Ramage, 2007).

One disadvantage of using WordNet for evaluating semantic relatedness is its hierarchical taxonomic structure. This results in terms such as *car* and *bicycle* being close in the network, but terms such as *car* and *gasoline* being far apart. Another difficulty arises from the non-scalability of WordNet. While the quality of the network is high, the manual nature of its construction means that arbitrary word pairs may not occur in the network. Hence in this paper we pursue an approach in which the resource for measuring semantic relatedness is created automatically, based on naturally occurring text.

A similar project, not using WordNet is WikiRelate (Strube and Ponzetto, 2006), which uses the existing link structure of Wikipedia as its base network, and uses similar path based measurements to those found in WordNet approaches to compute semantic relatedness. This project has seen improved results over most WordNet base approaches, largely due to the nature of Wikipedia, where articles tend to link to other articles which are related, rather than just ones which are similar.

2.2 Distributional Methods

An alternative method for judging semantic distance is using word co-occurrence statistics derived from a very large corpus (McDonald and Brew, 2004; Padó and Lapata, 2007) or from the web using search engine results (Turney, 2001).

In a recent paper, Agirre et al. (2009) parsed 4 billion documents (1.6 Terawords) crawled from the web, and then used a search function to extract syntactic relations and context windows surrounding key words. These were then used as features for vector space, in a similar manner to work done in (Padó and Lapata, 2007), using the British National Corpus (BNC). This system has produced excellent results, indicating that the quality of the results for these types of approaches is related to the size and coverage of their corpus. This does however present problems moving forward, as 1.6 Terawords is obviously an extremely large corpus, and it is likely that there would be a diminishing return on investment for increasingly large corpora. In the same paper, another method was shown which used the pagerank algorihm, run over a network formed from WordNet and the WordNet gloss tags in order to produce equally impressive results.

3 A Semantic Network Approach

The resource we use is a semantic network, automatically created by the large scale network creation program, ASKNet. The relations between nodes in the network are based on the relations returned by a parser and semantic analyser, which are typically the arguments of predicates found in the text. Hence terms in the network are related by the chain of syntactic/semantic relations which connect the terms in documents, making the network ideal for measuring the general notion of semantic relatedness.

Distinct occurrences of terms and entities are combined into a single node using a novel form of spreading activation (Collins and Loftus, 1975). This combining of distinct mentions produces a cohesive connected network, allowing terms and entities to be related across sentences and even larger units such as documents. Once the network is built, spreading activation is used to determine semantic

357

relatedness between terms. For example, to determine how related *car* and *gasoline* are, activation is given to one of the nodes, say *car*, and the network is "fired" to allow the activation to spread to the rest of the network. The amount of activation received by *gasoline* is then a measure of the strength of the semantic relation between the two terms.

We use three datasets derived from human judgements of semantic relatedness to test our technique. Since the datasets contain general terms which may not appear in an existing corpus, we create our own corpus by harvesting text from the web via Google. This approach has the advantage of requiring little human intervention and being extensible to new datasets. Our results using the semantic network derived from the web-based corpus are comparable to the best performing existing methods tested on the same datasets.

4 Creating the Semantic Networks

ASKNet creates the semantic networks using existing NLP tools to extract syntactic and semantic information from text. This information is then combined using a modified version of the update algorithm used by Harrington and Clark (2007) to create an integrated large-scale network. By mapping together concepts and objects that relate to the same real-world entities, the system is able to transform the output of various NLP tools into a single network, producing semantic resources which are more than the sum of their parts. Combining information from multiple sources results in a representation which would not have been possible to obtain from analysing the original sources separately.

The NLP tools used by ASKNet are the C&C parser (Clark and Curran, 2007) and the semantic analysis program Boxer (Bos et al., 2004), which operates on the CCG derivations output by the parser to produce a first-order representation. The named entity recognizer of Curran and Clark (2003) is also used to recognize the standard set of MUC entities, including `person`, `location` and `organisation`.

As an example of the usefulness of information integration, consider the *monk-asylum* example, taken from the RG dataset (described in Section 5.1). It is possible that even a large corpus could contain sentences linking *monk* with *church*, and linking *church* with *asylum*, but no direct links between *monk* and *asylum*. However, with an integrated semantic network, activation can travel across multiple links, and through multiple paths, and will show a relationship, albeit probably not a very strong one, between *monk* and *asylum*, which corresponds nicely with our intuition.

Figure 1, which gives an example network built from DUC documents describing the Elian Gonzalez custody battle, gives an indication of the kind of network that ASKNet builds. This figure does not give the full network, which is too large to show in a single figure, but shows the "core" of the network, where the core is determined using the technique described in (Harrington and Clark, 2009). The black boxes represent named entities mentioned in the text, which may have been mentioned a number of times across documents, and possibly using different names (e.g. Fidel Castro vs. President Castro). The diamonds are named directed edges, which represent relationships between entities.

A manual evaluation using human judges has been performed to measure the accuracy of ASKNet networks. On a collection of DUC documents, the "cores" of the resulting networks were judged to be 80% accurate on average, where accuracy was measured for the merged entity nodes in the networks and the relations between those entities (Harrington and Clark, 2009). The motivation for fully automatic creation is that very large networks, containing millions of edges, can be created in a matter of hours.

Automatically creating networks does result in lower precision than manual creation, but this is offset by the scalability and speed of creation. The experiments described in this paper are a good test of the automatic creation methodology.
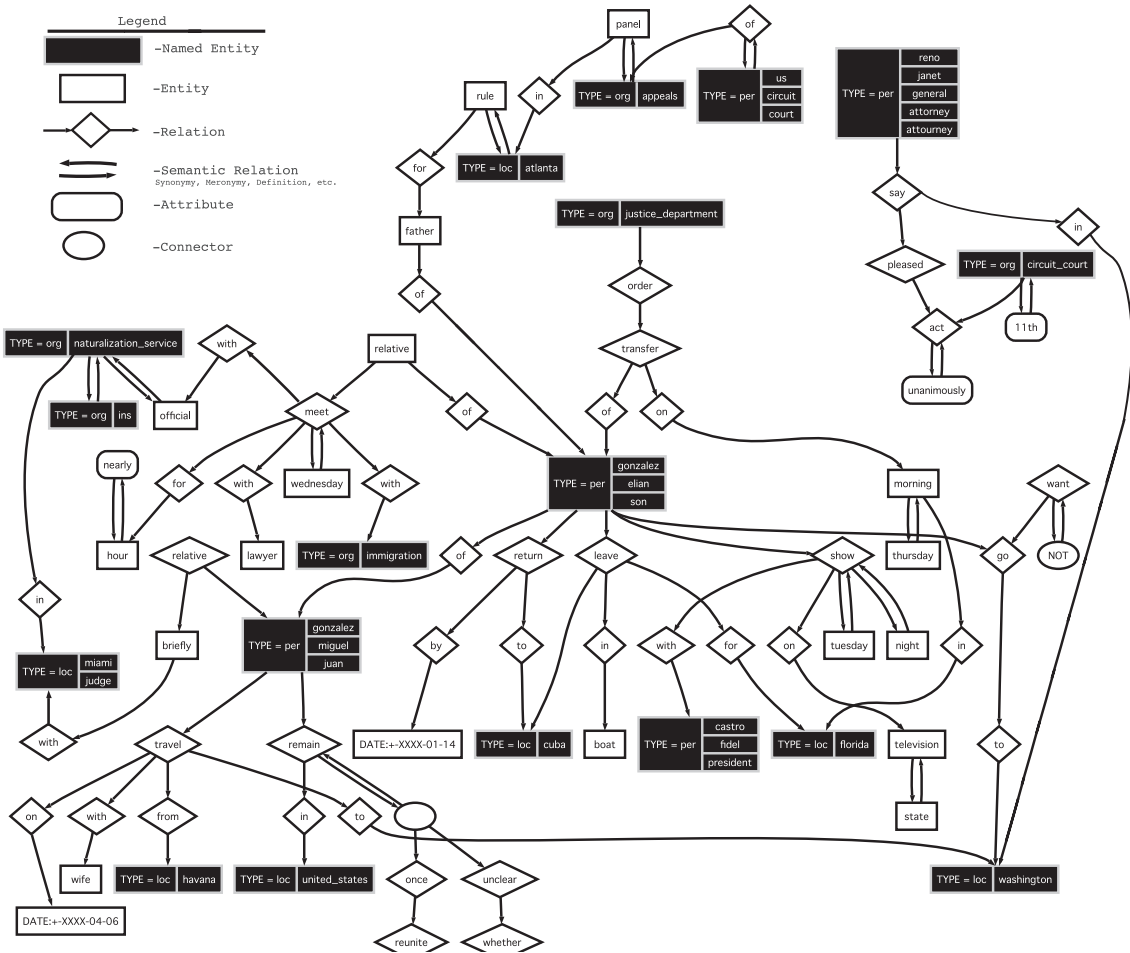
Figure 1: Example network core derived from DUC documents describing a news story from 2000.

The update algorithm

In order to merge information from multiple sources into a single cohesive resource, ASKNet uses a spreading activation based update algorithm (Harrington and Clark, 2007). As the system encounters new information, it attempts to map named entities in the new sentences it encounters to those already in its network. Each new sentence is turned into an *update fragment*; a fragment of the network representing the information contained in the sentence. Initial mapping scores, based on lexical similarity and named entity type, are made between the update fragment's named entities and those in the main network. The job of the update algorithm is to improve upon those initial scorings using the semantic information contained within the network.

The update algorithm iterates over each named entity node in the update fragment. This *base* node is provided with activation, which is allowed to spread throughout the fragment. All named entities which receive activation in this process, then transfer their activation to their *target* named entity nodes (nodes in the main network with which they have a current mapping score greater than zero). The amount transferred is based on the strength of the mapping score. The activation is then allowed to circulate through the main network until it reaches a stable state. At this point, the base node's mappings are updated based on which of its target nodes received activation. The more activation a target node receives, the more its mapping score with the

base node will increase.

The intuition behind the update algorithm is that we can use relatedness of nodes in the update fragment to determine appropriate mappings in the main network. So if our base node has the label "Crosby", and is related to named entity nodes referring to "Canada", "Vancouver" and "2010", those nodes will pass their activation onto their main network targets, and hopefully onto the node representing the ice hockey player Sidney Crosby. We would then increase the mapping score between our base node and this target, while at the same time decreasing the mapping score between out base node and the singer Bing Crosby, who (hopefully) would have received little or no activation. The update algorithm is also self-reinforcing, as in the successive stages, the improved scores will focus the activation further. In our example, in successive iterations, more of the activation coming to the "Crosby" node will be sent to the appropriate target node, and therefore there will be less spurious activation in the network to create noise.

For the purposes of these experiments, we extended the update algorithm to map together general object nodes, rather than focusing solely on named entities. This was necessary due to the nature of the task. Simply merging named entities would not be sufficient, as many of the words in datasets would not likely be associated strongly with any particular named entities. Extending the algorithm in this way resulted in a much higher frequency of mapping, and a much more connected final network. Because of this, we found that several of the parameters had to be changed from those used in Harrington and Clark (2009). Our initial activation input was set to double that used in Harrington and Clark's experiments (100 instead of 50), to compensate for the activation lost over the higher number of links. We also found that the number of iterations required to reach a stable state had increased to more than 4 times the previous number. This was to be expected due to the increased number of links

passing activation. We also had to remove the named entity type calculation from the initial mapping score, thus leaving the initial scoring to be simply the ratio of labels in the two nodes which overlapped. These changes were all done after manual observation of test networks built from searches not relating to any dataset, and were not changed once the experiments had begun.

4.1 Measuring semantic relatedness

Once a large-scale network has been constructed from a corpus of documents, spreading activation can be used to efficiently obtain a distance score between any two nodes in the network, which will represent the semantic relatedness of the pair. Each node in the network has a current amount of activation and a threshold (similar to classical ideas from the neural network literature). If a node's activation exceeds its threshold, it will fire, sending activation to all of its neighbours, which may cause them to fire, and so on. The amount of activation sent between nodes decreases with distance, so that the effect of the original firing is localized. The localized nature of the algorithm is important because it means that semantic relatedness scores can be calculated efficiently even for pairs of nodes in very large networks.

To obtain a score between nodes `x` and `y`, first a set amount of activation is placed in node `x`; then the network is fired until it stabilises, and the total amount of activation received by node `y` is stored as `act(x,y)`. This process is repeated starting with node `y` to obtain `act(y,x)`. The sum of these two values, which we call `dist(x,y)`, is used as the measure of semantic relatedness between `x` and `y`.[1]

`dist(x,y)` is a measure of the total strength of connection between nodes `x` and `y`, relative to the other nodes in their region. This takes into account not just direct paths, but also indirect paths, if the links along those paths are of sufficient strength. Since the

---

[1]The average could be used also but this has no effect on the ranking statistics used in the later experiments.

networks potentially contain a wide variety of relations between terms, the calculation of `dist(x,y)` has access to a wide variety of information linking the two terms. If we consider the *(car, gasoline)* example mentioned earlier, the intuition behind our approach is that these two terms are likely to be closely related in a semantic network built from text, either fairly directly because they appear in the same sentence or document, or indirectly because they are related to the same entities.

5 Experiments

The purpose of the experiments was to develop an entirely automated approach for replicating human judgements of semantic relatedness of words. We used three existing datasets of human judgements: the Hodgson, Rubenstein & Goodenough (RG) and Wordsimilarity-353 (WS-353) datasets. For each dataset we created a corpus using results returned by Google when queried for each word independently (Described in Section 5.2). We then built a semantic network from that corpus and used the spreading activation technique described in the previous section to measure semantic relatedness between the word pairs in the dataset.

The parser and semantic analysis tool used to create the networks were developed on newspaper data (a CCG version of the Penn Treebank (Steedman and Hockenmaier, 2007; Clark and Curran, 2007)), but our impression from informally inspecting the parser output was that the accuracy on the web data was reasonable. The experimental results show that the resulting networks were of high enough quality to closely replicate human judgements.

5.1 The datasets

Many studies have shown a marked priming effect for semantically related words. In his single-word lexical priming study, (Hodgson, 1991) showed that the presentation of a *prime word* such as *election* directly facilitates processing of a target word such as *vote*. Hodgson showed an increase in both re-sponse speed and accuracy when the prime and target are semantically related. 143 word pairs were tested across 6 different lexical relations: antonymy (e.g., *enemy, friend*); conceptual association (e.g., *bed, sleep*); category coordination (e.g., *train, truck*); phrasal association (e.g., *private, property*); superordination/subordination (e.g., *travel, drive*); and synonymy (e.g., *value, worth*). It was shown that equivalent priming effects (i.e., reduced processing time) were present across all relation types, thus indicating that priming was a result of the terms' semantic relatedness, not merely their similarity or other simpler relation type.

The Hodgson dataset consists of the 143 word pairs divided by lexical category. There were no scores given as all pairs were shown to have relatively similar priming effects. No examples of unrelated pairs are given in the dataset. We therefore used the unrelated pairs created by McDonald and Brew (2004).

The task in this experiment was to obtain scores for all pairs, and to do an ANOVA test to determine if there is a significant difference between the scores for related and unrelated pairs.

The WS-353 dataset (Finkelstein et al., 2002) contains human rankings of the semantic distance between pairs of terms. Although the name may imply that the scores are based on similarity, human judges were asked to score 353 pairs of words for their *relatedness* on a scale of 1 to 10, and so the dataset is ideal for our purposes. For example, the pair *(money, bank)* is in the dataset and receives a high relatedness score of 8.50, even though the terms are not lexically similar.

The dataset contains regular nouns and named entities, as well as at least one term which does not appear in WordNet (*Maradona*). In this experiment, we calculated scores for all word pairs, and then used rank correlation to compare the similarity of our generated scores to those obtained from human judgements.

The RG dataset (Rubenstein and Goodenough, 1965) is very similar to the WS-353,

though with only 65 word pairs, except that the human judges were asked to judge the pairs based on synonymy, rather than overall relatedness. Thus, for example, the pair *(monk,asylum)*, receives a significantly lower score than the pair *(crane,implement)*.

5.2 Data collection, preparation and processing

In order to create a corpus from which to build the semantic networks, we first extracted each individual word from the pairings, resulting in a list of 440 words for the ws-353 collection, 48 words for the RG (some words were used in multiple pairings), and 282 words for the Hodgson collection. For each of the words in this list, we then performed a query using Google, and downloaded the first 5 page results for that query. The choice of 5 as the number of documents to download for each word was based on a combination of informal intuition about the precision and recall of search engines, as well as the practical issue of obtaining a corpus that could be processed in reasonable space and time.

Each of the downloaded web pages was then cleaned by a set of Perl scripts which removed all HTML markup. Statistics for the resulting corpora are given in Table 1.

Three rules were added to the retrieval process to deal with problems encountered in formatting of web-pages:

1. Pages from which no text could be retrieved were ignored and replaced with the next result.
2. HTML lists preceded by a colon were recombined into sentences.
3. For Wikipedia disambiguation pages (pages which consist of a list of links to articles relating to the various possible senses of a word), all of the listed links were followed and the resulting pages added to the corpus.

Each of these heuristics was performed automatically and without human intervention.

The largest of the networks, created for the ws-353 dataset, took slightly over 24 hours

corpus	sentences	words
Hodgson	814,779	3,745,870
RG	150,165	573,148
ws-353	1,042,128	5,027,947

Table 1: Summary statistics for the corpora generated for the experiments.

to complete, including time for parsing and semantic analysis.

6 Results

6.1 Hodgson priming dataset

After processing the Hodgson corpus to build a semantic network with approximately 500,000 nodes and 1,300,000 edges, the appropriate node pairs were fired to obtain the distance measure as previously described. Those measurements were then recorded as measurements of semantic relatedness between two terms. If a term was used as a label in two or more nodes, all nodes were tried, and the highest scoring pairs were used.

As the Hodgson dataset did not provide examples of unrelated pairs against which we could compare, unrelated pairs were generated as described in (McDonald and Brew, 2004). This is not an ideal method, as several pairs that were identified as unrelated did have some relatively obvious relationship (e.g. *tree – house*, *poker – heart*). However we chose to retain the methodology for consistency with previous literature as it was also used in (Padó and Lapata, 2007).

Scores were obtained from the network for the word pairs, and for each target an average score was calculated for all primes in its category. Example scores are given in Table 2.

Two-way analysis of variance (ANOVA) was carried out on the network scores with the the relatedness status of the pair being the independent variable. A reliable effect was observed for the network scores with the primes for related words being significantly larger than those for unrelated words. The results are given in Table 3.

The use of ANOVA shows that there is a

Word pair	Related	Network Score
empty - full	Yes	10.13
coffee - mug	Yes	5.86
horse - goat	Yes	0.96
dog - leash	Yes	4.70
friend - antonym	No	0.53
vote - conceptual	No	1.37
property - phrasal	No	2.47
drive - super/sub	No	1.86

Table 2: Example scores obtained from the network for related and unrelated word pairs from the Hodgson dataset

	F	MSE	p	η^2
McDonald & Brew	71.73	0.004	< 0.001	
Padó & Lapata	182.46	0.93	< 0.01	0.332
PMI	42.53	3.79	< 0.001	0.263
Network	50.71	3.28	< 0.0001	0.411

Table 3: ANOVA results of scores generated from the Hodgson dataset compared to those reported for existing systems. (F = F-test statistic, MSE = Mean squared error, p = P-value, η^2 = Effect size)

6.2 ws-353 and rg datasets

The methodology used to obtain scores for the WS-353 and RG collections was identical to that used for the Hodgson data, except that scores were only obtained for those pairs listed in the data set. Because both collections provided direct scores, there was no need to retrieve network scores for unrelated pairings.

difference in the scores of the related and unrelated word pairs that cannot be accounted for by random variance. However, in order to compare the strength of the experimental effects between two experiments, additional statistics must be used. Eta-squared (η^2) is a measure of the strength of an experimental effect. A high η^2 indicates that the independent variable accounts for more of the variability, and thus indicates a stronger experimental effect. In our experiments, we found an η^2 of 0.411, which means that approximately 41% of the overall variance can be explained by the relatedness scores.

For comparison, we provide the ANOVA results for experiments by (McDonald and Brew, 2004) and (Padó and Lapata, 2007) on the same dataset. Both of these experiments obtained scores using vector based models populated with data from the BNC.

We also include the results obtained from performing the same ANOVA tests on Pointwise Mutual Information scores collected over our corpus. These results were intended to provide a baseline when using the web-based corpus. To calculate the PMI scores for this experiment, we computed scores for the number of times the two words appeared in the same paragraph or document, and the total number of occurrences of words in the corpus. The PMI scores were calculated by simply dividing the number of times the words co-occurred within a paragraph, by the product of the number of occurrences of each word within a document.

	WS-353	RG
WikiRelate!	0.48	0.86
Hughes-Ramage	0.55	0.84
Agirre Et Al	0.66	0.89
PMI	0.41	0.80
Network	0.62	0.86

Table 4: Rank correlation scores for the semantic network and PMI-based approaches, calculated on the WS-353 and RG collections, shown against scores for existing systems.

For consistency with previous literature, the scores obtained by the semantic network were compared with those from the collections using Spearman's rank correlation. The correlation results are given in Table 4. For comparison, we have included the results of the same correlation on scores from three top scoring systems using the approaches described above. We also include the scores obtained by using a simple PMI calculation as in the previous experiment.

The scores obtained by our system were not an improvement on those obtained by existing systems. However, our scores were on par with the best performing systems, which were purpose built for this application, and at least in the case of the system by Agirre et al. used a corpus several orders of magnitude larger.

7 Conclusion

In this paper we have shown that a semantic network approach to determining semantic relatedness of terms can achieve performance on par with the best purpose built systems. This is interesting for two reasons. Firstly, the approach we have taken in this paper is much more analogous to the way humans perform similar tasks. Secondly, the system used was not purpose built for this application. It is instead a general tool for information collection and integration, and this result indicates that it will likely be useful for a wide variety of language processing applications.

References

Agirre, Eneko, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of NAACL-HLT*.

Banerjee, Satanjeev and T. Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence*, Acapulco, Mexico.

Bos, Johan, Stephen Clark, Mark Steedman, James R. Curran, and Julia Hockenmaier. 2004. Wide-coverage semantic representations from a CCG parser. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-04)*, pages 1240–1246, Geneva, Switzerland.

Budanitsky, Alexander and Graeme Hirst. 2006. Evaluating wordnet-based measures of semantic distance. *Computational Linguistics*, 32:13 – 47, March.

Clark, Stephen and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Collins, Allan M. and Elizabeth F. Loftus. 1975. A spreading-activation theory of semantic processing. *Psychological Review*, 82(6):407–428.

Curran, James R. and Stephen Clark. 2003. Language independent NER using a maximum entropy tagger. In *Proceedings of the Seventh Conference on Natural Language Learning*, pages 164–167, Edmonton, Canada.

Fellbaum, Christiane, editor. 1998. *WordNet : An Electronic Lexical Database*. MIT Press, Cambridge, Mass, USA.

Finkelstein, Lev, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. In *ACM Transactions on Information Systems*, volume 20(1), pages 116–131.

Harrington, Brian and Stephen Clark. 2007. Asknet: Automated semantic knowledge network. In *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI'07)*, pages 889–894, Vancouver, Canada.

Harrington, Brian and Stephen Clark. 2009. Asknet: Creating and evaluating large scale integrated semantic networks. *International Journal of Semantic Computing*, 2(3):343–364.

Hodgson, James. 1991. Information constraints on pre-lexical priming. *Language and Cognitive Processes*, 6:169 – 205.

Hughes, Thad and Daniel Ramage. 2007. Lexical semantic relatedness with random graph walks. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 581–589, Prague, Czech Republic.

Jiang, J. J. and D. W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *International Conference on Research on Computational Linguistics*, Taipei, Taiwan, September.

McDonald, Scott and Chris Brew. 2004. A distributional model of semantic context effects in lexical processing. In *Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics*, pages 17 – 24, Barcelona, Spain.

Mohammad, Saif and Graeme Hirst. 2006. Distributional measures of concept-distance: A task-oriented evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, Sydney, Australia.

Padó, Sebastian and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Resnik, Philip. 1999. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130.

Rubenstein, H. and J.B. Goodenough. 1965. Contextual correlates of synonymy. *Computational Linguistics*, 8:627 – 633.

Steedman, Mark and Julia Hockenmaier. 2007. Ccgbank: A corpus of ccg derivations and dependency structures extracted from the penn treebank. *Computational Linguistics*, 33:355–396.

Strube, Michael and Simone Paolo Ponzetto. 2006. Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings of the 21st national conference on Artificial intelligence*, pages 1419–1424. AAAI Press.

Turney, Peter D. 2001. Lecture notes in computer science 1: Mining the web for synonyms: PMI-IR versus LSA on TOEFL.