Unsupervised All-words Word Sense Disambiguation with Grammatical Dependencies

Vivi Nastase
EML Research gGmbH
Heidelberg, 69118, Germany
nastase@eml-research.de

Abstract

We present experiments that analyze the necessity of using a highly interconnected word/sense graph for unsupervised all-words word sense disambiguation. We show that allowing only grammatically related words to influence each other's senses leads to disambiguation results on a par with the best graph-based systems, while greatly reducing the computation load. We also compare two methods for computing selectional preferences between the senses of every two grammatically related words: one using a Lesk-based measure on WordNet, the other using dependency relations from the British National Corpus. The best configuration uses the syntactically-constrained graph, selectional preferences computed from the corpus and a PageRank tie-breaking algorithm. We especially note good performance when disambiguating verbs with grammatically constrained links.

1 Introduction

It has long been believed that being able to detect the correct sense of a word in a given context – performing word sense disambiguation (WSD) – will lead to improved performance of systems tackling high end applications such as machine translation (Chan et al., 2007) and summarization(Elhadad et al., 1997). In order for WSD methods to be useful, they must be robust, portable, scalable, and therefore preferably not reliant on manually tagged data. These desiderata have lead to an increased interest in developing unsupervised WSD methods, flexible relative to the word sense inventory, and which disambiguate all open-class words in a given context as opposed to a selected few.

Particularly appropriate from this point of view are graph-based methods (Navigli and Lapata, 2007), which map the open-class words in a given context onto a highly interconnected graph. Each node in this graph represents a word sense, and weighted edges will connect every pair of senses (corresponding to different words). The topology of the graph and the weights of the edges can contribute in a variety of ways to determine the best sense combination for the words in the considered context. This approach leads to large and highly interconnected graphs, in which distant, unrelated (in the context) words, are nonetheless connected, and allowed to influence each other's sense preferences. We study the impact on disambiguation performance when connections are restricted to pairs of word senses corresponding to words that are grammatically linked in the considered context.

The benefits of using grammatical information for automatic WSD were first explored by Yarowsky (1995) and Resnik (1996), in unsupervised approaches to disambiguating single words in context.

Sussna (1993) presents a first approach to disambiguating together words within a context. The focus is on nouns, and the sense combination that minimizes the overall distance in the WordNet nouns network is chosen.

Stetina et al. (1998) present the first approach, supervised, to disambiguating all words in a sentence with sense association (or selectional) preferences computed from a sense-tagged corpus. An untagged grammatically linked word pair will have associated a matrix of sense combination scores, based on the analyzed sense-tagged corpus, and similarities between the current words and those in tagged pairs with the same grammatical relation. Once such matrices are computed for all grammatically related word pairs, the sense preferences are propagated from the bottom of the parse tree towards the top, and the sense selection starts from the top and propagates downward.

McCarthy and Carroll (2003) also use an unsupervised approach and grammatical relations to learn selectional preferences for word classes. In an approach inspired by the works of Li and Abe (1998) and Clark and Weir (2002), McCarthy and Carroll use grammatically connected words from a corpus to induce a distribution of senses over subtrees in the WordNet hierarchy. McCarthy et al. (2004) use a corpus and word similarities to induce a ranking of word senses from an untagged corpus to be used in WSD.

We build upon this previous research, and propose an unsupervised WSD method in which senses for two grammatically related words in the sentence will be connected through directed edges. We experiment with graph edge weights computed using WordNet, and weights computed using grammatical collocation information from a corpus. These

weights are used to induce an initial scoring of the graph vertices, starting from the leaves and propagating upwards. The disambiguation process starts with choosing a sense for the head of the sentence, and moves towards the leaves, propagating downward the chosen senses at each step, and using the edge weights and vertex scores to guide the sense selection process.

We investigate two issues: (i) whether using in disambiguation only syntactically connected words leads to results on a par with, or better than, using all word-sense combinations, (ii) whether sense association strength induced from a sense-unlabeled corpus can rival relatedness measures induced from a lexical resource - in our case, WordNet.

We evaluate this approach on the Senseval-2(Palmer et al., 2001) and Senseval-3(Snyder and Palmer, 2004) English all-words test data. On the Senseval-2 data we obtain results on a par with the best unsupervised WSD systems, on the Senseval-3 data, the results are lower overall, but for verbs higher than those obtained with other graph-based methods. In both situations, using only grammatically motivated edges leads to improved disambiguation of verbs compared to disambiguating in a graph with unrestricted connections.

2 Disambiguation Algorithm

The disambiguation method described here uses grammatical information from the sentential context to constrain word pairs that are allowed to influence each other's sense choice. Edge weights in the graph are relatedness scores computed based on WordNet and, in a second set-up, selectional preferences estimated from an (sense-)untagged corpus, for disambiguating together all words in the sentence. Grammatical information for the sentential context is obtained using the dependency relation output of the Stanford Parser (de Marneffe et al., 2006). Selectional preferences are estimated using grammatical collocation information from the British National Corpus (BNC), obtained with the Word Sketch Engine (WSE) (Kilgarriff et al., 2004).

2.1 Extracting grammatical relation information

We parse the Senseval test data using the Stanford Parser(Klein and Manning, 2003) generating the output in dependency relation format (de Marneffe et al., 2006). Edges that do not connect open-class words are filtered out, words are lemmatized, and we reintroduce the copula (it is bypassed as a predicate) because the verb *be* must be disambiguated as well.

To estimate selectional preferences from a sense-untagged corpus, for each grammatically related pair of words in a sentence we extract evidence consisting of pairs with the same grammatical relation and either the same head or dependent, using the Word Sketch Engine. To obtain such pairs we map the grammatical relations used by the Stanford Parser onto the set of grammatical relations used by the WSE. Table 1 shows a sample of this mapping. We denote by GR^{-1} the inverse of grammatical relation GR – for example *subject_of* is the inverse of *subject*.

Dependency relation	WSE relation
nsubj(verb,noun)	subject(verb,noun) subject_of(noun,verb)
dobj(verb,noun)	object(verb,noun) object_of(noun,verb)
amod(noun,adj)	a_modifier(noun,adj) modifies(adj,noun)
nn($noun_1$,$noun_2$)	n_modifier($noun_1$,$noun_2$) modifies($noun_2$,$noun_1$)
prep_of(verb,noun)	pp_of(verb,noun) pp-obj_of(noun,verb)

Table 1: Mapping of grammatical relations from the Stanford Parser onto the WSE relation set – a sample.

The result of this processing is illustrated in Figure 1, for the following sentence from the Senseval2 test data:

The art of change-ringing is peculiar to the English, and, like most English peculiarities, unintelligible to the rest of the world.

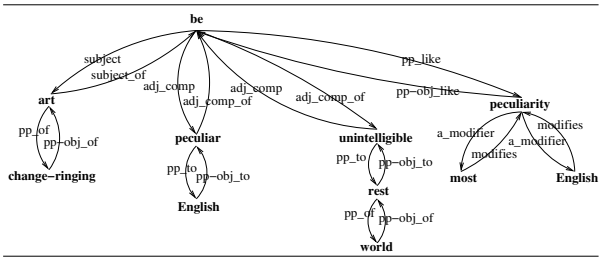

Figure 1: Dependency graph with grammatical relations mapped onto the WSE set

The dependency between two connected words is represented by two asymmetric grammatical relations.

2.2 Computing sense selectional preference scores

The selectional preference scores can be computed using the lexical resource that provides the inventory of senses, or using a corpus.

Sense-selectional preferences based on dependency relations in a corpus For each pair of words in a grammatical relation (w_1, w_2, GR) from a sentence, we compute a score for each sense s^i_{w2} of w_2, that shows the strength of the association

between s^i_{w2} and w_1. The strength of the association will come from collocation information from the BNC, combined with sense similarity or relatedness between $s^i_{w_2}$ and collocates of w_1 in grammatical relation GR.

Let us take an example – *(rest,world,pp_of)* from the example sentence presented before. We want to estimate the preferences of *rest* for senses of *world*. *world* has the following senses in WordNet 1.7[1]:
world%1:17:02::[2], world%1:17:00::, world%1:17:01::, world%1:14:02::, world%1:14:01::, world%1:14:00::, world%1:09:01::, world%1:09:00:: .

From the BNC we obtain the following collocation information (the formatting of the list is w_1-POS GR w_x-POS:co-occurrence frequency):

rest-n	pp_of	life-n:639, world-n:518, Europe-n:211, cast-n:44, season-n:90, day-n:253, country-n:158, family-n:134, evening-n:60, Kingdom-n:42, chapter-n:55, team-n:96, week-n:93, society-n:89, afternoon-n:34, population-n:56, ...

The list of grammatical collocates with $rest$ in relation *pp_of* are: $GC^{pp\_of}_{rest}$ = { *life, world, Europe, case, season, day, country, family, evening, Kingdom, chapter, team, week, society, afternoon, population,...* }

Based on relatedness scores between senses of these collocates and senses of *world* we compute selectional preference scores for each of *world*'s senses:

world%1:17:02::→1	world%1:17:00::→2
world%1:17:01::→3	world%1:14:02::→2
world%1:14:01::→3	world%1:14:00::→4
world%1:09:01::→1	world%1:09:00::→1

The same procedure is applied to compute the sense selectional preference scores of $world$ for each of $rest$'s senses, in the grammatical relation *pp-obj_of* (the inverse of *pp_of* in WSE).

Formally, for the tuple (w_1, w_2, GR), we extract from the BNC all pairs (w_1, w_x, GR)[3]. The set

$$GC^{GR}_{w_1} = \{w_x | (w_1, w_x, GR) \in \text{corpus}\}$$

gives w_1's grammatical collocations. To estimate the sense association strength between w_1 and senses of w_2, for each $w_x \in GC^{GR}_{w_1}$ we compute relatedness between the senses of w_x and the senses of w_2. $A^{s^i_{w_2}}_{w_1|GR}$, the association strength between w_1 and sense $s^i_{w_2}$ of word w_2 under relation GR, is the sum of these relatedness scores:

$$A^{s^i_{w_2}}_{w_1|GR} = \sum_{w_x \in GC^{GR}_{w_1}} \sum_{s^j_{w_x} \in S_{w_x}} rel(s^i_{w_2}, s^j_{w_x})$$

where S_{w_x} is the set of senses for word w_x.

If this value is 0, then $A^{s^i_{w_2}}_{w_1|GR} = \frac{1}{n_{w_2}}$, where n_{w_2} is the number of senses of w_2.

$rel(s^i_{w_2}, s^j_{w_x})$ can be computed as a similarity or relatedness measure (Budanitsky and Hirst, 2006). Because the sense inventory for the Senseval data comes from WordNet and we work at the sense level, we use relatedness measures based on WordNet, as opposed to corpus-based ones. In the experiments presented further in the paper, we have used a relatedness measure based on hypernym and hyponym information, in the following manner:

$$rel(s^i_{w_2}, s^j_{w_x}) = \begin{cases} 1 & : \ s^i_{w_2} \text{ is a hypernym of } s^j_{w_x} \\ 1 & : \ s^i_{w_2} \text{ is a hyponym of } s^j_{w_x} \\ & \text{and } path\_length(s^i_{w_2}, s^j_{w_x}) \leq 2 \\ 1 & : \ s^i_{w_2} \text{ similar to/antonym of } s^j_{w_x} \\ 0 & : \ otherwise \end{cases}$$

In other words, if the sense $s^i_{w_2}$ of w_2 is a hypernym of the sense $s^j_{w_x}$ or a close hyponym (distance at most 2) or connected through a *similar to/antonym of* relation, we consider the two senses related and relatedness gets a score of 1. Otherwise, we consider the two senses unrelated.

The motivation for using this relatedness measure is that it allows fast computations – essential when dealing with a large amount of information from a corpus – and it clusters closely related senses based on WordNet's hypernym/hyponym relations. By clustering together related senses, we gather more evidence for the selectional preferences of w_2's senses, which also helps partly with the data sparseness problem.

Because at this point it is not determined to which of w_x's senses the selectional preference is due, all of w_x's senses will have the same selectional preference to a sense j of w_y: $A^{s^j_{w_y}}_{s^i_{w_x}|GR} = A^{s^j_{w_y}}_{w_x|GR}$, for all senses $s^i_{w_x}$ of w_x.

Sense-selectional preferences based on a lexical resource When using the lexical resource, because we have pairs that connect words under different parts of speech, we opt for a Lesk-based measure (Banerjee and Pedersen, 2003). Relatedness scores are computed for each pair of senses of the grammatically linked pair of words (w_1, w_2, GR), using the WordNet-Similarity-1.03 package and the `lesk`

[1] WordNet 1.7 is the sense inventory for Senseval2, WordNet 1.7.1 is the sense inventory for Senseval 3.

[2] Unique sense identifier from the WN lexicographer files.

[3] Only w_x collocates that have the same part of speech as w_2 are considered.

option (Pedersen et al., 2004). To maintain the notation from above, we denote by $A^{s^i_{w_x}}_{s^j_{w_y}}$ the lesk relatedness score between sense i of w_x and sense j of w_y. These scores are symmetric: $A^{s^i_{w_x}}_{s^j_{w_y}} = A^{s^j_{w_y}}_{s^i_{w_x}}$, and independent of grammatical relations GR.

2.3 The sense-enhanced dependency tree

After computing the sense association strength scores for w_1 and w_2 in grammatical relation GR in the sentence, we expand the edge (w_x, w_y, GR) from the dependency tree to the two sets of directed edges:

$$\{(s^i_{w_x} \rightarrow s^j_{w_y}, GR) | i = 1, n; j = 1, m\},$$
$$\{(s^j_{w_y} \rightarrow s^i_{w_x}, GR^{-1}) | i = 1, n; j = 1, m\}.$$

The weight of an edge $(s^i_{w_x} \rightarrow s^j_{w_y}, GR)$ is $A^{s^j_{w_y}}_{s^i_{w_x}|GR}$. Figure 2 shows one sense-enhanced edge.

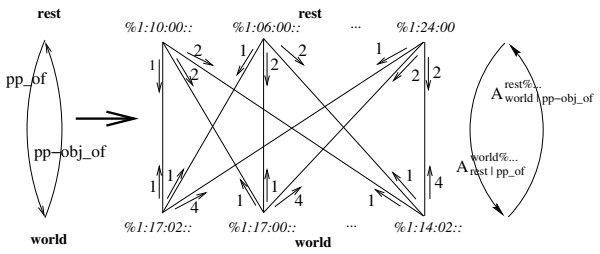

Figure 2: A sense enhanced edge, with weights induced from corpus collocations.

2.4 Word sense disambiguation

We first compute a score for each vertex (word sense) using the estimated sense preferences, traversing the dependency graph from the bottom up[4]. Each leaf is given a score of $\frac{1}{n_w}$, where n_w is the number of senses of the word w to which the leaf pertains. The score of the other vertices are the weighted sum of the scores of their grammatical dependents in the sentence under analysis:

$$Score(s^i_{w_x}) = \sum_{(w_x, w_y, GR)} \sum_{s^j_{w_y} \in S_{w_y}} A^{s^j_{w_y}}_{s^i_{w_x}|GR} \times Score(s^j_{w_y})$$

The word sense disambiguation process starts from the root node of the dependency tree. The highest ranked score for the root is chosen, and the nodes corresponding to the other senses and their edges are deleted from the graph. For each of its dependents we add the sense preferences imposed by the chosen sense to the vertex's score, and proceed with the sense selection in the same way down through the graph.

$$Score(s^j_{w_y}) = Score(s^j_{w_y}) + A^{s^j_{w_y}}_{s^i_{w_x}|GR^{-1}}$$

where (w_x, w_y, GR) $((w_y, w_x, GR^{-1}))$ is in the current sentence.

Because of data sparseness, there may be not enough evidence in the corpus to produce a clear winner, and several senses are tied. All senses are then kept, and disambiguation proceeds further. If more than one word has multiple senses left after the top-down traversal of the tree, we use two methods: random choosing from the tied senses or the sequence labeling method described in (Mihalcea, 2005). The graph's vertices are the senses that remain to be disambiguated, and its edges connect every pair of these senses (provided that they correspond to different words). The score of each vertex is initially set to 1, and the edge weights are Lesk similarity scores. The vertices are scored using a Page Rank algorithm, in which the rank at every iteration step is computed with the formula:

$$WP(a) = (1 - d) + d \sum_{b \in In(a)} \frac{w_{ba}}{\sum_{c \in Out(b)} w_{bc}} WP(b)$$

where:
a, b, c are vertices in the graph;
$WP(a)$ is the weighted PageRank score of node a;
d is the probability that there will be a jump from a given vertex to another in the graph. We use $d = 0.85$, the value set by (Brin and Page, 1998) for Google's PageRank model.
$In(a)$ is the set of a's predecessors;
$Out(a)$ is the set of a's successors.

When the vertex scores converge[5], the highest ranking vertex for each word will give the sense prediction for that word.

For multi-term expressions that are split during parsing (such as *come back*), for which there is no prediction since they do not appear as such in the parse tree, the system randomly picks one of the WordNet senses.

3 Experiments and Results

The WSD algorithm proposed is evaluated on the Senseval-2 and Senseval-3 English-all-words task test data. Table 2 shows the results obtained for fine-grained scoring. Because for each target there is a prediction, precision and recall have the same value.

[4]The up-down orientation of the graph is given by the dependency tree from which it was expanded.

[5]An aperiodic, irreducible graph is guaranteed to converge (Grimmett and Stirzaker, 1989). For every graph we built that has more than 3 nodes, the aperiodicity condition is met – it has cycles of length 2 and 3, therefore the greatest common divisor of its cycle lengths is 1. The graph is also irreducible – it has no leaves because it is highly interconnected.

POS	$Rand.$	$Seq.$	GR_{WN}	GR^{PR}_{WN}	GR_{BNC}	GR^{PR}_{BNC}
Senseval 2						
noun	41.1%	63.0%	58.9%	62.4%	54.2%	**63.3%**
verb	22.0%	31.6%	31.0%	**33.0%**	30.9%	32.7%
adjective	38.9%	**56.8%**	52.9%	**56.8%**	40.4%	**56.8%**
adverb	53.2%	57.5%	53.2%	58.8%	53.2%	**59.1%**
all	36.7%	52.1%	49.0%	52.4%	44.6%	**52.7%**
Senseval 3						
noun	42.5%	58.2%	53.2%	55.4%	40.3%	**58.6%**
verb	19.4%	40.4%	40.3%	**42.3%**	19.9%	40.0%
adjective	45.0%	56.7%	53.4%	54.5%	46.0%	**57.5%**
adverb	92.9%	92.9%	92.9%	92.9%	92.9%	92.9%
all	34.4%	50.8%	48.2%	50.1%	33.8%	**51.2%**

Table 2: Precision (= Recall) disambiguation results for Senseval English-all-words test data

Column $Random$ ($Rand.$) shows a simple random baseline, and column $Sequence$ ($Seq.$) shows the sequence data labelling method (Mihalcea, 2005) – one of the best performing graph-methods (Navigli and Lapata, 2007). The results presented were obtained using word similarities computed with the WordNet-Similarity-1.03 package, on a sense graph built using the marked targets in the test set. These results are not the same as those reported in (Mihalcea, 2005) for the Senseval 2 data (nouns 57.5%, verbs: 36.5%, adjective: 56.7%, adverb: 70.9%, for an average precision of 54.2%), because of the difference in computing word similarities. The other 4 columns show results obtained using grammatical relation information between words as identified by the parser. GR_{WN} includes the results obtained using the Lesk-based similarity with the syntactically-based graph and breaking ties randomly, GR^{PR}_{WN} presents results obtained in a similar configuration – only the tie breaking is done using PageRank. GR_{BNC} and GR^{PR}_{BNC} are similar with the previous two columns, only in this case the edge weights are the selectional preference scores induced from the BNC.

The performance of GR_{WN} is close to that of Seq. When ties are broken randomly, the computation is much faster, since we do two traversals of a small graph, while PageRank iterates until convergence (approx. 15 iterations) on graphs of average size of 1500 edges and 52 vertices (on Senseval 2 data). When PageRank is used to solve ties the performance on GR^{PR}_{WN} surpasses that of Seq while still being faster, having to iterate over graphs with an average of 1074 edges and 40 vertices. The computation load is not only lighter during disambiguation, but also in the data preparation stage, when similarities must be computed between every sense pair corresponding to every pair of words within a sentence (or a window of a given size).

There are other important differences. While the syntactic structure of the sentence plays no role in the $Sequence$ method, it is crucial for the other methods. In the Senseval data not all words in a sentence were tagged as targets, and the $Sequence$ method works only on them. This is not the case for the GR methods, which work with the full syntactic tree – and will disambiguate more words at a time. Also, the targets tagged in the data contain "satellites" information (e.g. *turn out*, *set up*), which may change the part of speech of the main target (e.g. *at the same time* (adv) for target *time* (noun), *out of print* (adj) for target *print* (noun)). Multi-word expressions are themselves the subject of ample research, and we could not incorporate them into our corpus-based approach. Verb particles in particular pose a problem, as most parsers will interpret the particle as a preposition or adverb. This was the case for the Senseval data, as well. On the other hand, this is a more realistic set-up, with no reliance on previously marked targets.

Selectional preferences induced from a corpus without sense annotations perform well for verbs, but overall do not perform very well by themselves. The reasons for this are multiple. The most important is data sparseness. Many sense selection preferences are 0. In order to improve this approach, we will look into more flexible methods for computing dependency pair similarities (without fixing one of the vertices as we did in this paper). Previous research in inducing sense rankings from an untagged corpus (McCarthy et al., 2004), and inducing selectional preferences at the word level (for other applications) (Erk, 2007) will provide the starting point for research in this direction.

4 Comparison with Related Work

The most similar approach to the one we describe, that has been tested on Senseval-2, is the one described in (McCarthy and Carroll, 2003). The best results reported are 51.1% precision and 23.2% recall. This implementation also used grammatical information and selectional preferences induced from a corpus to determine a disjoint partition – determined by a cut in the WordNet is-a tree – over which it computes a probability distribution conditioned by the grammatical context and a verb or adjective class.

McCarthy et al. (2004) report a disambiguation precision of 53.0% and recall of 49.0% on the Senseval-2 test data, using an approach that derives sense ranking based on word similarity and distributional analysis in a corpus.

Mihalcea (2005) reports the highest results on the Senseval-2 data obtained with a graph-based algorithm – 54.2% precision and recall. The results obtained with a PageRank algorithm applied to a sense graph built from a words within a context of a given

size are also the highest for a completely unsupervised WSD[6] system in Senseval-2.

The best result obtained by an unsupervised system on the Senseval-3 data is reported by Strapparava et al. (2004) – 58.3%. This implementation uses WordNet-Domains, a version of WordNet enhanced with domain information (e.g. economy, geography). The domain of a given text is automatically detected, and this information will constrain the possible senses of words in the given text.

For Senseval 3 data, using a graph method with the *Key Player Problem* to measure vertex relevance, Navigli and Lapata (2007) report very close results to (Strapparava et al., 2004) on nouns and adjectives, and lower scores for verbs (F1-scores: 61.9% for nouns, 62.8% for adjectives, 36.1% for verbs compared with 62.2% for nouns, 66.9% for adjectives, 50.4% for verbs). Mihalcea (2005) reports an overall score of 52.2% for this data.

It is interesting to look at the dependency tree we used for WSD from the point of view of graph connectivity measures (Navigli and Lapata, 2007). To determine the importance of a node in a graph, whether it represents the words and their senses in a given context, or people in a social network, one can use different measures. According to grammatical theories, the importance of a node in the sentence parse tree is given by the phrase type it heads, and the number of words it thus dominates. From this point of view, the top-down propagation of senses traverses and disambiguates the tree in order of the decreasing importance of nodes. Other methods could be used as well, such as disambiguating first the most highly connected nodes – the ones with the most sense constraints.

5 Conclusions

We have studied the impact of grammatical information for constraining and guiding the word sense disambiguation process in an unsupervised all-words setup. Compared with graph methods, the approach we described is computationally lighter, while performing at the same level on Senseval-2 and Senseval-3 all-words tasks test data. Grammatical constraints serve both to limit the number of word-senses pair similarities necessary, and also to estimate selectional preferences from an untagged corpus.

Using only grammatically motivated connections leads to better disambiguation of verbs for both Senseval-2 and Senseval-3 test data, but while the difference is consistent (1.4%, 1.9%) it is not statistically significant.

[6]As opposed to other unsupervised approaches, the sense frequency information from WordNet was not used.

We explored a new method for estimating sense association strength from a sense-untagged corpus. Disambiguation when using sense relatedness computed from WordNet is very close in performance with disambiguation based on sense association strength computed from the British National Corpus, and on a par with state-of-the-art unsupervised systems on Senseval-2. This indicates that grammatical relations and automatically derived sense association preference scores from a corpus have high potential for unsupervised all-word sense disambiguation.

References

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlap as a measure of semantic relatedness. In *Proc. of IJCAI-03,* Acapulco, Mexico, 9–15 August, 2003, pages 805–810.

Sergey Brin and Larry Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30:1–7.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based measures of semantic distance. *Computational Linguistics*, 32(1):13–47.

Yee Seng Chan, Hwee Tou Ng, and David Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proc. of ACL-07*, pages 33–40.

Stephen Clark and David Weir. 2002. Class-based probability estimation using a semantic hierarchy. *Computational Linguistics*, 28(2):187–206.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure. In *Proc. of LREC-06*.

Michael Elhadad, Kathleen R. McKeown, and Jaques Robin. 1997. Floating constraints in lexical choice. *Computational Linguistics*, 23(2):195–239.

Katrin Erk. 2007. A simple, similarity-based model for selectional preferences. In *Proc. of ACL-07*, pages 216–223.

Geoffrey Grimmett and David Stirzaker. 1989. *Probability and Random Processes*. Oxford University Press.

Adam Kilgarriff, Pavel Rychly, Pavel Smrz, and David Tugwell. 2004. The Sketch Engine. In *Proc. of EuraLex-04*, pages 105–116.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proc. of ACL-03*, pages 423–430.

Hang Li and Naoki Abe. 1998. Generalizing case frames using a thesaurus and the MDL principle. *Computational Linguistics*, 24(2):217–244.

Diana McCarthy and John Carroll. 2003. Disambiguating nouns, verbs and adjectives using automatically acquired selectional preferences. *Computational Linguistics*, 29(4):639–654.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant senses in untagged text. In *Proc. of ACL-04,* Barcelona, Spain, 21–26 July 2004, pages 280–287.

Rada Mihalcea. 2005. Large vocabulary unsupervised word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proc. of HLT-EMNLP-05*, pages 411–418.

Roberto Navigli and Mirella Lapata. 2007. Graph connectivity measures for unsupervised word sense disambiguation. In *Proc. of IJCAI-07*, pages 1683–1688.

Martha Palmer, Christiane Fellbaum, Scott Cotton, Lauren Delfs, and Hoa Trang Dang. 2001. English tasks: all-words and verb lexical sample. In *Proc. of the ACL SENSEVAL-2 Workshop, Toulouse, France, 2001*.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. WordNet::Similarity – Measuring the relatedness of concepts. In *Proc. of HLT-NAACL-04,* Boston, Mass., 2–7 May, 2004, pages 267–270.

Philip Resnik. 1996. Selectional constraints: an information-theoretic model and its computational realization. *Cognition*, (61):127–159, November.

Benjamin Snyder and Martha Palmer. 2004. The English all-words task. In *Proc. of the ACL SENSEVAL-3 Workshop, Barcelona, Spain, 2004*, pages 41–43.

Jiri Stetina, Sadao Kurohashi, and Makoto Nagao. 1998. General word sense disambiguation method based on a full sentential context. In Sanda Harabagiu, editor, *PROC. of Use of WordNet in Natural Language Processing Systems*, pages 1–8.

Carlo Strapparava, Alfio Gliozzo, and Claudio Giuliano. 2004. Pattern abstraction and term similarity for word sense disambiguation. In *Proc. of the ACL SENSEVAL-3 Workshop, Barcelona, Spain, 2004*, pages 229–234.

Michael Sussna. 1993. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proc. of the CIKM-93*, pages 67–74.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivalling supervised methods. In *Proc. of ACL-05,* Cambridge, Mass., 26–30 June 1995, pages 189–196.