

Evaluating a Semantic Network Automatically Constructed from Lexical Co-occurrence on a Word Sense Disambiguation Task

Sean Szumlanski
Department of EECS
University of Central Florida
seansz@cs.ucf.edu

Fernando Gomez
Department of EECS
University of Central Florida
gomez@eecs.ucf.edu

Abstract

We describe the extension and objective evaluation of a network¹ of semantically related noun senses (or *concepts*) that has been automatically acquired by analyzing lexical co-occurrence in Wikipedia. The acquisition process makes no use of the metadata or links that have been manually built into the encyclopedia, and nouns in the network are automatically disambiguated to their corresponding noun senses without supervision. For this task, we use the noun sense inventory of WordNet 3.0. Thus, this work can be conceived of as augmenting the WordNet noun ontology with unweighted, undirected *related-to* edges between synsets. Our network contains 208,832 such edges.

We evaluate our network's performance on a word sense disambiguation (WSD) task and show: a) the network is competitive with WordNet when used as a stand-alone knowledge source for two WSD algorithms; b) combining our network with WordNet achieves disambiguation results that exceed the performance of either resource individually; and c) our network outperforms a similar resource that has been automatically derived from semantic annotations in the Wikipedia corpus.

1 Introduction

A growing interest in using semantic relatedness in word sense disambiguation (WSD) tasks has spurred investigations into the limitations of the WordNet ontology (Fellbaum, 1998) for this purpose. Although WordNet comprises a rich set of semantic

links between word senses (or *concepts*), indicating semantic similarity through subsumptive hypernymic and hyponymic relations (among others), it lacks a general indication of semantic relatedness.

We present a semantic network that is automatically acquired from lexical co-occurrence in Wikipedia, and indicates general semantic relatedness between noun senses in WordNet 3.0. In this work, the discovery of relatedness is a context-sparse affair that takes place *in absentia* of the semantic annotations of Wikipedia, such as inter-article links, entries in disambiguation pages, the title of the article from which a sentence is extracted, and so on.

We released an earlier version of such a network that was limited by the fact that only relationships involving at least one monosemous noun had been included, and it was not evaluated on a WSD task (Szumlanski and Gomez, 2010).

In contrast, the network we present here has relatedness data for over 4,500 polysemous noun targets and 3,000 monosemous noun targets, each of which are related to an average of 27.5 distinct noun senses. It consists of 208,832 undirected edges – a 181% increase in size over the previous network. The result is a semantic network that has reached maturity and, as we will show, can be successfully applied to a WSD task.

This paper proceeds as follows. In the next section (Section 2), we discuss related work. We then give an overview of the method we use to construct our network (Sections 3 and 4). The network is evaluated through its application to a WSD task (Sections 5–7), where we compare its performance to WordNet and another automatically acquired semantic network called WordNet++ (Ponzetto and Navigli, 2010). A discussion follows (Section 8),

¹<http://www.cs.ucf.edu/~seansz/sem>

and we present our conclusions in Section 9.

2 Related Work

Our work bears strong relation to WordNet++ (henceforth WN++), which is constructed automatically from the semantic annotations in Wikipedia (Ponzetto and Navigli, 2010).² Links in WN++ are established between words whose articles link to one another. For example, the article on *astronomy* in Wikipedia links to the article on *celestial navigation*, so we find an edge from `astronomy#n#1` to `celestial_navigation#n#1` in WN++.³ The nouns related in WN++ are disambiguated automatically using further semantic annotation data from Wikipedia, including sense labels, the titles of other pages linked to by any two related nouns, and the folksonomic categories to which articles belong. These serve as context words that are compared with context words from various WordNet relations in order to map the nouns to their appropriate WordNet senses. The resulting resource contains 1,902,859 unique edges between noun senses.

Augmenting the structure of Wikipedia itself has been the subject of research as well, and involves the discovery of relations between articles. Mihalcea and Csomai (2007), for example, added links between Wikipedia pages after automatically identifying keywords in each article and disambiguating those words to their appropriate Wikipedia concepts (article titles), while Ponzetto and Navigli (2009) used graph theoretic approaches to augment the taxonomic organization of Wikipedia articles.

In terms of automatically discovering semantic relations, many pattern-based approaches have been used to extract specific types of relations from large corpora, e.g., hyponymy, meronymy, and synonymy (Hearst, 1992; Pantel and Pennacchiotti, 2006). Approaches based on distributional similarity have been applied toward the same end (Harris, 1985; Gorman and Curran, 2006), and there are several approaches that rely on the underlying structure of WordNet or Wikipedia to measure the relatedness between two concepts or nouns quantitatively (Hughes and Ramage, 2007; Gabrilovich and

²<http://lcl.uniroma1.it/wordnetplusplus>

³The notation `astronomy#n#1` refers to sense 1 (#1) of the noun (#n) “astronomy” in WordNet. Other parts of speech are denoted by #v (verbs), #a (adjectives), or #r (adverbs).

Markovitch, 2007; Zaragoza et al., 2007; Patwardhan and Pedersen, 2006; Strube and Ponzetto, 2006; Budanitsky and Hirst, 2006; Resnik, 1995).

Other quantitative approaches have leveraged the large amounts of data available on the Web to discover relatedness. Notably, Agirre and de Lacalle (2004) employed web queries to associate WordNet synsets with representative context words, known as topic signatures. Cuadros and Rigau (2008) have used these data to construct four KnowNets, semantic knowledge bases derived by disambiguating the top 5, 10, 15, and 20 nouns, respectively, from the topic signatures of Agirre and de Lacalle.

3 Automatic Acquisition of the Semantic Network

The semantic network is automatically acquired in three distinct stages (Szumlanski and Gomez, 2010): (1) quantitative measurement of relatedness between nouns that co-occur in a large corpus; (2) categorical determination of whether the quantitative measure indicates strong and mutual semantic relatedness between a given pair of nouns; and (3) unsupervised disambiguation of all the nouns that are found to be semantically related. We provide an overview of each of these steps below (Sections 3.1–3.3), and then discuss how we have expanded this methodology to create a more complete semantic network (Section 4).

3.1 Quantitatively measuring relatedness from lexical co-occurrence

We first measure the semantic relatedness, or **relational strength**, of a target, t , to one of its co-occurring nouns, or co-targets, c , with the following asymmetric function:

$$S_{rel}(t, c) = P(t|c)P(c|t)\log\frac{P(c|t)}{P(c)}$$

where $P(c|t)$ is the relative frequency of c among all nouns co-occurring with t , and vice versa for $P(t|c)$. $P(c)$ is the relative frequency of c among all nouns occurring in the corpus. For these values, we rely on lexical co-occurrence data extracted from Wikipedia. Co-occurrence is considered intra-sententially (as opposed to co-occurrence in entire articles or paragraphs, or co-occurrence within variable-sized windows of context).

This function essentially measures the degree to which an occurrence of t in a sentence predicts the co-occurrence of c . It is an adaptation of Resnik’s (1999) selectional association measure.

Table 1 shows the results of applying this function to the co-targets of “yoga” and “meditation.”

Target (t): yoga		Target (t): meditation	
Co-target (c)	S_{rel}	Co-target (c)	S_{rel}
hatha yoga	.1801	yoga	.0707
asana	.0761	mindfulness	.0415
meditation	.0673	contemplation	.0165
bhakti	.0508	prayer	.0139
raja	.0410	practice	.0068
tantra	.0148	technique	.0060
yogi	.0132	mantra	.0053
karma	.0125	relaxation	.0048
posture	.0104	retreat	.0047
aerobics	.0093	enlightenment	.0031
tai chi	.0089	monk	.0025
exercise	.0036	posture	.0024
practice	.0032	breathing	.0017
instructor	.0031	-----	----
-----	----	exercise	.0015
guru	.0027	teaching	.0014
massage	.0026	practitioner	.0014
exercise	.0019	ascetic	.0014
⋮	⋮	⋮	⋮

Table 1: The most strongly related co-targets of “yoga” and “meditation,” sorted by decreasing value of relational strength (S_{rel}). Nouns above dashed lines are the top 5% of the target’s most strongly related co-targets.

3.2 Establishing categorical relatedness

We then use a mutual relatedness algorithm to ascertain whether two nouns are semantically related by determining whether the nouns under consideration reciprocate a high degree of relatedness to one another. It proceeds as follows:

For some target noun of interest, t , let $C_x(t)$ be the set of the top $x\%$ of t ’s co-targets as sorted by $S_{rel}(t, c)$. For each $c \in C_x(t)$, if we have $t \in C_x(c)$, then we say that t and c are categorically related and add the noun pair (t, c) to our semantic network. We then increment x by one and repeat the process: for every $c \in C_{x+1}(t)$ such that (t, c) is not already in our network, we look for t in $C_{x+1}(c)$, and add (t, c)

to our network if we find it. This process continues until we have incremented x some number of times without adding any new relations to the semantic network. We then take the symmetric closure of the network, so that if (t, c) is in the network, (c, t) is, as well. (That is, the relation is considered undirected.)

Consider, for example, the nouns in Table 1. Given the target “yoga,” we might first examine the top 5% of its most strongly related co-targets (an arbitrary initial threshold chosen simply for illustrative purposes). In this case, we have all the nouns above the dashed line: $C_5(yoga) = \{hatha\ yoga, asana, meditation, bhakti, raja, tantra, yogi, karma, posture, aerobics, tai\ chi, exercise, practice, instructor\}$. The algorithm then searches $C_5(hatha\ yoga)$, $C_5(asana)$, and so on, for “yoga,” adding a new relation to the network every time “yoga” is found. Thus, we can see by the inclusion of “yoga” in $C_5(meditation)$ (all nouns above the dashed line in the second column of Table 1), that the pair $(yoga, meditation)$ will be included in the network.

This reliance on mutual relatedness ensures that only noun pairs exhibiting strong semantic relatedness are admitted to the network.

3.3 Disambiguation

Disambiguation of the resulting noun-noun pairs is the product of majority-rules voting by the following three algorithms.

Subsumption. The most frequently occurring immediate hypernyms of all nouns related to our target are permitted to disambiguate the polysemous nouns. This is useful because of the semantic clustering that tends to occur among related nouns. (E.g., “astronomer” is related to several terms categorized as celestial bodies in WordNet, such as “planet,” “star,” “minor planet,” and “quasar.”)

Glosses. Senses of polysemous co-targets with occurrences of monosemous co-targets in their glosses are preferentially taken as the intended meanings of the polysemous nouns. Monosemous co-targets are matched directly, or by suffix replacement. (E.g., “biology” can be matched by the occurrence of “biologist” in a gloss, “engineering” by “engineers,” and so on.)

Selectional Preferences. This method associates a numerical score with all superordinate synsets from the WordNet noun ontology that categorize

the monosemous nouns related to a target. For example, the noun “unicorn” strongly predicts related nouns categorized as monsters (monster#1)⁴ and mythical beings (mythical_being#1) in WordNet. These selectional preferences are applied to polysemous co-targets in decreasing order of their relational strength to the target noun. A polysemous noun is disambiguated to the first sense or senses subsumed by one of these selectional preferences. For example, “phoenix,” as it relates to “unicorn,” is disambiguated to phoenix#3 in WordNet (the fiery bird that is reborn from its own ashes) by virtue of its subsumption by mythical_being#1.

4 Creating a More Complete Network

A shortcoming of our previously released network is that it lacked concept-level relations between pairs of polysemous nouns.

When humans encounter a pair of ambiguous but closely related words, like *bus-horn*, we automatically disambiguate to the automobile and the car horn, as opposed to a computer’s front-side bus or a rhinoceros’s horn. The human ability to perform this disambiguation stems from the fact that human semantic memory relates not just individual words, but specific concepts denoted by those words. But if our goal is to establish such a link in our computational model of semantic relatedness, then we cannot rely on the link to perform that disambiguation for us; another approach is called for.

One reasonable approach (the one taken in our previous work) is to go where the problem no longer exists – to relationships that involve at least one monosemous noun. Monosemous-to-monosemous noun relationships require no disambiguation. Monosemous-to-polysemous noun relationships, on the other hand, require that only one noun be disambiguated. This ameliorates our problem tremendously, because the monosemous noun in the pair anchors the polysemous noun in an unambiguous context where disambiguation can more readily take place. That context includes all the nouns related to our monosemous noun, which, through their transitive relatedness to the polysemous noun in question, can assist in the act of disambiguation *vis-à-vis* the algorithms described in Section 3.3.

Consider, in contrast, the polysemous “batter,” which can refer to the baseball player or the cake batter. The algorithm for discovering semantic relatedness yields several nouns related to each of these senses of “batter” (see Table 2). If we wish to disambiguate the pair (batter, cake), we are left with the question: which of the nouns in Table 2 should we take as contextual anchors for the disambiguation?

baking	fastball	inning	strike
ball	flour	outfielder	strikeout
base	glove	pancake	swing
baseball	hitter	pitch	tempura
bat	home plate	pudding	umpire
cake	home run	runner	waffle
dugout	infielder	shortstop	

Table 2: An excerpt of some of the nouns related to “batter” by the algorithm for automatic acquisition.

In considering this question, it is important to note that although the ontological categories that subsume the nouns related to “batter” exhibit greater entropy than we usually observe among the terms related to a monosemous noun, clear delineations still exist. For example, Figure 1 shows the clusters that form as we consider shared hypernymic relationships between all senses of the nouns related to “batter” (gray nodes in the graph). We see that many of the nouns related to “batter” have senses categorized by food#1, cake#3, pitch#2, ballplayer#1, or equipment#1 – the heads of five distinct clusters by semantic similarity.

It is worth noting that some nouns related to “batter” (such as “baking,” “swing,” and “umpire”) do not fall into any of these semantic clusters. In these cases, the WordNet glosses serve as our primary tool for disambiguation. (For example, the glosses of both swing#8 and umpire#1 include mention of “baseball,” which is also related to “batter.”)

Conversely, some of the polysemous nouns in our example have senses that join semantic clusters unintendedly. For instance, cake#2 (“[a] small flat mass of chopped food,” according to WordNet) falls under the cluster headed by food#1. Although this is potentially problematic, cake#2 is discarded in this particular case in favor of cake#3 (the baked good),

⁴We sometimes drop the part of speech from our word sense notation for brevity, but only in the case of noun senses.

which has a greater mass because of its subsumption of waffle#1 and pancake#1, and is indeed the intended meaning of “cake” as it relates to “batter.”

Another example of unintended cluster membership comes from bat#4 (the cricket bat), which is categorized by sports_equipment#1. In contrast, the baseball bat does not have its own entry in WordNet, and the most reasonable sense choice, bat#5 (“a club used for hitting a ball in various games”), is categorized as a stick (stick#1), and not as equipment, sports equipment, or game equipment.

These unintended cluster memberships are bound to cause minor errors in our disambiguation efforts. However, our analysis reveals that we do not find such high entropy among the relatives of a polysemous noun that the semantic clustering effect (which is necessary for the success of the disambiguation algorithms described above in Section 3.3) is diminished. Thus, to construct our network, we apply the disambiguation algorithms described above, with the following modification: when confronted with a pair of semantically related polysemous nouns, we apply the disambiguation mechanism described above in both directions, and then fuse the results together. So, in one direction, the various baked goods related to “batter” help us to properly disambiguate “cake” to cake#3 in WordNet, yielding the pair (batter, cake#3). A similar scenario yields the pair (cake, batter#2) when disambiguating in the other direction, and we fuse the results together into the properly disambiguated pair (batter#2, cake#3).

Using this method, we have automatically created a semantic network that has 208,832 pairs of related noun senses – the most extensive semantic network between WordNet noun senses to be derived automatically from a simple lexical co-occurrence measure. For the remainder of this paper, we will refer to our network as the Szumlanski-Gomez network (SGN).

5 Coarse-Grained WSD Experiments

To evaluate our semantic network, and to provide fair comparison to related work, we take our cue from Ponzetto and Navigli (2010), who evaluated the performance of WN++ on the SemEval-2007 (Navigli et al., 2007) coarse-grained all-words WSD task using extended gloss overlaps (Banerjee and

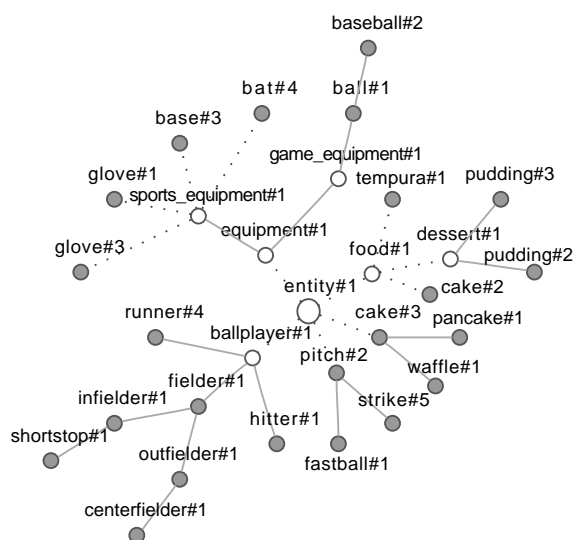


Figure 1: A partial view of the WordNet graph, showing senses of nouns related to “batter” (gray nodes) and intermediary concepts (white nodes) that connect them to the root of the taxonomy through hypernymic relationships.

Pedersen, 2003) and the graph-based degree centrality algorithm (Navigli and Lapata, 2010).

In this particular SemEval task, we are presented with 237 sentences in which lemmatized target words have been flagged for disambiguation. In our experiments, we disambiguate nouns only (as did Ponzetto and Navigli), since both SGN (our network) and WN++ relate only concepts denoted by nouns, and no other parts of speech. In our experimental setup, each sentence is considered in isolation from the rest, and all lemmatized content words in a sentence are provided to the disambiguation algorithms; the verbs, adjectives, and adverbs, although we do not resolve their senses, lend additional context to the disambiguation algorithms.

The coarse-grained nature of the SemEval-2007 task provides that there may be more than one acceptable sense assignment for many of the targets. In the coarse-grained setting, an algorithm’s sense assignment is considered correct when it appears in the list of acceptable senses for the given target word.

The algorithms below both allow for multiple disambiguation results to be returned in the event of a tie. In these cases (although they are rare), we adopt the approach of Banerjee and Pedersen (2003), who award partial credit and discredit proportionally for all the senses returned by the algorithm.

6 Extended Gloss Overlaps (ExtLesk)

The first disambiguation algorithm we employ is the extended gloss overlaps measure (henceforth ExtLesk) of Banerjee and Pedersen (2003), which is an extension of the Lesk (1986) gloss overlap measure. Loosely speaking, the algorithm disambiguates a target noun by maximizing the overlap (number of words in common) between the glosses of word senses related⁵ to the target’s noun senses and those related to all context words (all lemmatized targets in the sentence under consideration other than the target itself). The sense with the greatest overlap is selected as the intended meaning of a target noun.

In the event of a tie, multiple senses may be selected. ExtLesk does not attempt to perform sense assignment if the score for every sense of a target noun is zero, except when dealing with a monosemous noun, in which case we default to the only sense possible.

6.1 Results

We have run ExtLesk on the SemEval-2007 task using five combinations of semantic resources: WordNet only, SGN (our semantic network) only, SGN and WordNet combined (that is, the union of all links contained in both networks), WN++ only, and WN++ combined with WordNet. We include the traditional baselines of most frequent sense (MFS) assignment and random sense assignment for comparison, and measure precision (number of correct sense assignments divided by the number of attempted sense assignments), recall (number of correct sense assignments divided by the number of target nouns to be disambiguated), and the harmonic mean of the two, F_1 , defined as:

$$F_1 = \frac{2 * precision * recall}{precision + recall}$$

We present our results in Table 3, and offer the following observations. Firstly, SGN as a stand-alone network rivals the performance of WordNet. This is particularly impressive given the fact that

⁵We use all relations available in WordNet, as well as a *related-to* relation derived from the links in our semantic network.

Resource	P	R	F ₁
WordNet	78.80	74.82	76.76
SGN	78.64	72.82	75.62
SGN and WordNet	82.35	78.11	80.18
WN++	74.67	61.87	67.67
WN++ and WordNet	77.35	73.38	75.31
MFS Baseline	77.40	77.40	77.40
Random Baseline	63.50	63.50	63.50

Table 3: ExtLesk disambiguation results on the SemEval-2007 all-words coarse-grained WSD task (nouns only).

the edges in SGN were derived automatically from a simple lexical co-occurrence measure.

Equally impressive is the ability of SGN and WordNet, when used in combination, to achieve results that exceed what either network is able to accomplish as a stand-alone knowledge source. When combined, we see improvements of 3.42% and 4.56% over WordNet and SGN as stand-alone resources, respectively. It is also only with these resources combined that we are able to outperform the MFS baseline, and we do so by 2.78%.⁶

In contrast, WN++ fails to perform as a stand-alone resource, falling behind the MFS baseline by 9.73%.⁷ Of all the resources tested, WN++ yields the lowest results. When combined with WordNet, WN++ actually diminishes the ability of WordNet to perform on this WSD task by 1.45%. We defer our discussion of factors impacting the performance of WN++ to Section 8 (Discussion).

7 WSD with Degree Centrality

Degree centrality is a graph-based measure of semantic relatedness (Navigli and Lapata, 2010) in which we search through a semantic network for paths of length $l \leq maxLength$ between all sense nodes for all lemmas in our context. The edges along all such paths are added to a new graph, G' , and for each target noun to be disambiguated, the sense node with the greatest number of incident edges (highest vertex degree) in G' is taken as its intended sense.

⁶Other systems have obtained better results on the same dataset, but we focus only on SGN and WN++ because our aim is to compare the resources themselves.

⁷Ponzetto and Navigli (2010) report results of $F_1 = 68.3$ and 72.0 for WN and WN++ as stand-alone resources. Space considerations prevent us from discussing this disparity in detail.

In these graphs, nodes represent synsets, as opposed to instantiating separate nodes for different members of the same synset and allowing edges to be constructed between them. We include all lemmas from a sentence in our context, but only return disambiguation results for the nouns.

With SGN and WN++, the implementation of this algorithm is straightforward. We initiate a breadth-first search (BFS)⁸ at each target sense node in the network, and proceed through $\lfloor \frac{maxLength+1}{2} \rfloor$ iterations of spreading activation. Whenever the tendrils of this spreading activation from one target sense node in the graph connect to those of another,⁹ we add the path between the nodes to our new graph, G' , potentially incrementing the degree of the involved target sense nodes in G' as we do so.

Because BFS is an admissible algorithm (guaranteed to find the shortest path from an initial state to a goal), it provides a computationally efficient approach to finding all paths between all target nodes. Also, because any node on a path of length $l \leq maxLength$ between two target nodes is at most $\lfloor \frac{l}{2} \rfloor$ nodes removed from at least one of those target sense nodes, we only need to perform a BFS of depth $\lfloor \frac{maxLength+1}{2} \rfloor$ from every target sense node in order to guarantee that every such path between them will be discovered. Since the time complexity of BFS is exponential with respect to the depth of the search, cutting this depth in half (in comparison to performing a BFS of depth $maxLength$) greatly reduces the running time of our algorithm.

We take the same approach in traversing the WordNet noun graph, using all possible sense relations as edges. In keeping with the approach of Navigli and Lapata (2010), an edge is also induced between synsets if the gloss of one synset contains a monosemous content word. For example, the gloss for `leprechaun#n#1`, “a mischievous elf in Irish folklore;” contains the monosemous noun “folklore;” thus, we have an edge between `leprechaun#n#1` and

⁸This is in contrast to the DFS implementation of Navigli and Lapata (2010), so for the sake of posterity, we expound upon our approach in this section.

⁹When $maxLength$ is odd, this requires an additional check to ensure that the intersection is not taking place at a node that is exactly $\lfloor \frac{maxLength+1}{2} \rfloor$ degrees removed from each of the two target nodes it is connecting, as this would result in a path with overall length $maxLength + 1$ between the target nodes.

`folklore#n#1` in the WordNet graph.

Once we have our new graph, G' , constructed in this manner, the vertex degree is considered an indication of the semantic relatedness of a particular synset to all other lemmas in our context. For each target noun, we use its sense node with the highest degree in G' for sense assignment.

7.1 Results

We have tested the degree centrality algorithm with the following combinations of semantic resources: WordNet, SGN, WN++, Refined WN++, SGN and WordNet combined, and Refined WN++ and WordNet combined. (Refined WN++ consists of 79,422 of WN++’s strongest relations, and was created in an unsupervised setting by Ponzetto and Navigli specifically for use with degree centrality when they discovered that WN++ had too many weak relations to perform well with the algorithm.)

We have observed that the performance of degree centrality rapidly levels off as $maxLength$ increases. Ponzetto and Lapata (2010) also report this so-called “plateau” effect, and employ a $maxLength$ of 6 in their experiments, despite finding that results level off around $maxLength = 4$. We, too, find that performance levels off around $maxLength = 4$ in almost all cases, and so only continue up to $maxLength = 5$.

We find that, in all cases tested, degree centrality is unable to outperform the MFS baseline (with respect to F_1) (see Table 4). SGN and WN++ exhibit comparable performance with this algorithm, with maximum F_1 values of 68.4% ($maxLength = 2$) and 67.3% ($maxLength = 3-5$), respectively. Neither achieves the performance of WordNet with degree centrality ($F_1 = 74.0\%$), which underperforms the MFS baseline ($F_1 = 77.4\%$) by 3.4%.¹⁰ Ponzetto and Navigli (2010) reported that only performing sense assignment when the max degree exceeded an empirically derived but non-disclosed threshold improved performance, but we have found that implementing such a threshold universally lowers results for all resources we tested with degree centrality.

¹⁰Although Ponzetto and Navigli (2010) reported similar results with WordNet ($F_1 = 74.5$), we have been unable to reproduce their results using Refined WN++, either combined with WordNet ($F_1 = 79.4$) or as a stand-alone resource ($F_1 = 57.4$).

The lowest performance using degree centrality comes from Refined WN++ as a stand-alone resource. We attribute this to the fact that Refined WN++ is so semantically sparse. On average, noun senses in Refined WN++ are related to only 3.42 other noun senses, while those in WN++ and SGN relate to an average of 44.59 and 10.92 noun senses, respectively. Accordingly, the success of Refined WN++ and WordNet combined is attributable mostly to the success of WordNet as a stand-alone resource; as *maxLength* increases, the contributions made by the sparse Refined WN++ network rapidly become negligible in comparison to those provided by the WordNet ontology.

<i>l</i>	P	R	F ₁	P	R	F ₁
	WordNet			SGN		
1	96.9	16.8	28.6	79.7	32.9	46.6
2	77.6	45.1	57.0	72.0	64.6	68.4
3	76.7	65.6	70.7	68.7	63.5	66.0
4	76.9	71.0	73.9	68.0	63.9	65.9
5	76.6	71.6	74.0	68.0	64.2	66.1
	SGN & WN			WN++		
1	77.4	52.4	62.5	87.2	23.5	37.1
2	74.7	70.7	72.7	71.6	60.2	65.4
3	70.3	67.1	68.7	70.7	64.3	67.3
4	70.5	67.4	68.9	70.4	64.5	67.3
5	70.1	67.0	68.5	70.4	64.5	67.3
	WN _{refined} ⁺⁺			WN _{refined} ⁺⁺ & WN		
1	98.3	15.3	26.5	83.3	31.2	45.4
2	91.4	23.4	37.3	77.5	66.6	71.6
3	88.7	29.9	44.7	77.6	73.6	75.5
4	83.7	32.3	46.7	74.7	71.4	73.0
5	80.2	35.3	49.0	74.7	71.4	73.0
	MFS Baseline			Random Baseline		
	77.4	77.4	77.4	63.5	63.5	63.5

Table 4: Degree centrality disambiguation results on the SemEval-2007 all-words coarse-grained WSD task (nouns only). *l* is maximum path length.

8 Discussion

The fact that the performance of degree centrality quickly plateaus hints at the root cause of its weak performance compared to ExtLesk and the MFS baseline. As the maximum path length is increased in a dense semantic network, all possible edges from

our target sense nodes rapidly find themselves involved with paths to other target sense nodes. This is particularly true of WN++ (notice its rapid and stable convergence), where certain “sticky” nodes form bridges between seemingly unrelated concepts. For example, the frequent appearance of “United States” in Wikipedia articles, and its tendency to be linked to the *United States* Wikpage when it occurs, causes the term to serve as a bridge between such diverse concepts as *automaton#2* and *burrito#1*, which one would typically expect to be far removed from one another in a model of semantic relatedness.

Nonetheless, the degree centrality algorithm has no difficulty finding short paths between target sense nodes when traversing any of the semantic networks we tested. In fact, we have discovered that as the results of degree centrality converge, they approach the performance obtained by foregoing the algorithm altogether and simply disambiguating each noun to the sense with the most edges in the network (regardless of whether those edges ultimately connect two word senses from the disambiguation context). The expected values of convergence attained by defaulting to the most semantically well-connected sense of each target noun in each network are $F_1 = 66.3\%$, 67.5% , and 74.6% for SGN, WN++, and WordNet, respectively – remarkably close to the experimentally derived degree centrality results of $F_1 = 66.1\%$, 67.3% , and 74.0% .

9 Conclusion

We have constructed a semantic network of related noun senses automatically from intra-sentential lexical co-occurrence data, and shown that on a WSD task, it outperforms a similar resource, WN++, which is derived from the rich set of semantic annotations available in the Wikipedia corpus. Our network has also shown competitive performance with the WordNet ontology on WSD, and when combined with WordNet, improves disambiguation results in a coarse-grained setting using the ExtLesk disambiguation algorithm.

Acknowledgments

This research was supported in part by the NASA Engineering and Safety Center under Grant/Cooperative Agreement NNX08AJ98A.

References

- Eneko Agirre and Oier Lopez de Lacalle. 2004. Publicly available topic signatures for all WordNet nominal senses. In *Proceedings of the 4th International Conference on Language Resources and Evaluations (LREC '04)*, pages 1123–1126, Lisbon, Portugal.
- Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI '03)*, pages 805–810, Acapulco, Mexico.
- Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.
- Montse Cuadros and German Rigau. 2008. KnowNet: building a large net of knowledge from the web. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING '08)*, pages 161–168, Manchester, UK. Association for Computational Linguistics.
- Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
- Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI '07)*, pages 1606–1611, Hyderabad, India.
- James Gorman and James R. Curran. 2006. Scaling distributional similarity to large corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL '06)*, pages 361–368, Sydney, Australia. Association for Computational Linguistics.
- Zellig S. Harris. 1985. Distributional structure. In J. J. Katz, editor, *The Philosophy of Linguistics*, pages 26–47. Oxford University Press.
- Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING '92)*, pages 539–545, Nantes, France.
- Thad Hughes 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 (EMNLP-CoNLL '07)*, pages 581–589, Prague, Czech Republic. Association for Computational Linguistics.
- Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation (SIGDOC '86)*, pages 24–26, Toronto, Ontario, Canada. ACM.
- Rada Mihalcea and Andras Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM '07)*, pages 233–242, Lisbon, Portugal. ACM.
- Roberto Navigli and Mirella Lapata. 2010. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):678–692.
- Roberto Navigli, Kenneth C. Litkowski, and Orin Hargraves. 2007. SemEval-2007 Task 07: coarse-grained English all-words task. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval '07)*, pages 30–35, Prague, Czech Republic. Association for Computational Linguistics.
- Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL '06)*, pages 113–120, Sydney, Australia. Association for Computational Linguistics.
- Siddharth Patwardhan and Ted Pedersen. 2006. Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics Workshop on Making Sense of Sense*, pages 1–8, Trento, Italy.
- Simone Paolo Ponzetto and Roberto Navigli. 2009. Large-scale taxonomy mapping for restructuring and integrating Wikipedia. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI '09)*, pages 2083–2088, Pasadena, CA.
- Simone Paolo Ponzetto and Roberto Navigli. 2010. Knowledge-rich word sense disambiguation rivaling supervised systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*, pages 1522–1531, Uppsala, Sweden. Association for Computational Linguistics.
- Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI '95)*, pages 448–453, Montreal, QC.
- Philip Resnik. 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.

- Michael Strube and Simone Paolo Ponzetto. 2006. Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI '06)*, pages 1419–1424, Boston, MA. AAAI Press.
- Sean Szumlanski and Fernando Gomez. 2010. Automatically acquiring a semantic network of related concepts. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM '10)*, pages 19–28, Toronto, Ontario, Canada. ACM.
- Hugo Zaragoza, Henning Rode, Peter Mika, Jordi Atserias, Massimiliano Ciaramita, and Giuseppe Attardi. 2007. Ranking very many typed entities on Wikipedia. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM '07)*, pages 1015–1018, Lisbon, Portugal. ACM.