

Using Web Selectors for the Disambiguation of All Words

Hansen A. Schwartz and Fernando Gomez

School of Electrical Engineering and Computer Science

University of Central Florida

Orlando, FL 32816, USA

{hschwartz, gomez}@cs.ucf.edu

Abstract

This research examines a word sense disambiguation method using selectors acquired from the Web. Selectors describe words which may take the place of another given word within its local context. Work in using Web selectors for noun sense disambiguation is generalized into the disambiguation of verbs, adverbs, and adjectives as well. Additionally, this work incorporates previously ignored adverb context selectors and explores the effectiveness of each type of context selector according to its part of speech. Overall results for verb, adjective, and adverb disambiguation are well above a random baseline and slightly below the most frequent sense baseline, a point which noun sense disambiguation overcomes. Our experiments find that, for noun and verb sense disambiguation tasks, each type of context selector may assist target selectors in disambiguation. Finally, these experiments also help to draw insights about the future direction of similar research.

1 Introduction

The great amount of text on the Web has emerged as an unprecedented electronic source of natural language. Recently, word sense disambiguation systems have fostered the size of the Web in order to supplant the issue of limited annotated data availability for supervised systems (Mihalcea, 2002; Agirre and Martinez, 2004). Some unsupervised or minimally supervised methods use the Web more directly in disambiguation algorithms that do not use a training set for the specific target words.

One such minimally supervised method uses selectors acquired from the Web for noun sense disambiguation by comparing the selectors of a given sen-

tence to a target noun within the sentence (Schwartz and Gomez, 2008). Although this work found strong results, many aspects of the use of selectors was left unexplored. For one, the method was only applied to noun sense disambiguation, focusing on the well-developed noun hypernym hierarchy within WordNet (Miller et al., 1993). Additionally, the role of different types of selectors was not extensively explored, and adverb selectors were not used at all. We seek to address those issues.

In this paper, we extend our method of using selectors from the Web for noun sense disambiguation into a more robust method of disambiguating words of all parts of speech. After a brief background on selectors and related work, we explain the acquisition and empirical application of selectors from nouns, verbs, adjectives, pronouns/proper nouns, and adverbs. Finally, results are presented from the SemEval-2007 coarse grained all-words task (Navigli et al., 2007), and we explore the influence of various types of selectors on the algorithm in order to draw insight for future improvement of Web-based methods.

2 Background

In this section we describe related research in selectors and solving the problem of word sense disambiguation (WSD). Specifically, two types of WSD research are examined: works that used the Web in direct manner, and works which applied a similarity or relatedness measure.

2.1 Selectors

The term selector comes from (Lin, 1997), and refers to a word which can take the place of another given word within the same local context. Although

Lin searched a dependency relationship database in order to match local context, it is not yet possible to parse dependency relationships of the entire Web. In turn, one must search for text as local context. For example, in the sentence below, the local context for ‘strikers’ would be composed of “he addressed the” and “at the rally.”

He addressed the strikers at the rally.

Previously, we introduced the idea of using selectors of other words in a sentence in addition to selectors of the target, the word being disambiguated (Schwartz and Gomez, 2008). Words taking the place of a target word are referred to as *target selectors* and words which take the place of other words in a sentence are referred to as *context selectors*. *Context selectors* can be classified further based on their part of speech. In our example, if ‘striker’ was the target word, the *verb context selectors* would be verbs replacing ‘addressed’ and the *noun context selectors* would be nouns replacing ‘rally’.

Similarity is used to measure the relationship between a target word and its *target selectors*, while *relatedness* measures the relationship between a target word and *context selectors* from other parts of the sentence. Thus, the use of selectors in disambiguating words relies on a couple assumptions:

1. Concepts which appear in matching syntactic constructions are *similar*.
2. Concepts which appear in the context of a given target word are *related* to the correct sense of the target word.

Note that ‘concept’ and ‘word sense’ are used interchangeably throughout this paper. This idea of distinguishing similarity and relatedness has an extensive history (Rada et al., 1989; Resnik, 1999; Patwardhan et al., 2003; Budanitsky and Hirst, 2006), but most algorithms only find a use for one or the other.

2.2 Related Word Sense Disambiguation

A key aspect of using selectors for disambiguation is the inclusion of context in the Web search queries. This was done in works by (Martinez et al., 2006) and (Yuret, 2007), which substituted relatives or similar words in place of the target word within a

given context. The context, restricted with a window size, helped to limit the results from the Web. These works followed (Mihalcea and Moldovan, 1999; Agirre et al., 2001) in that queries were constructed through the use of a knowledge-base, filling the queries with pre-chosen words. We also use context in the web search, but we acquire words matching a wildcard in the search rather than incorporate a knowledge-base to construct queries with pre-chosen relatives. Consequently, the later half of our algorithm uses a knowledge-base through similarity and relatedness measures.

Some recent works have used similarity or relatedness measures to assist with WSD. Particularly, (Patwardhan et al., 2003) provide evaluations of various relatedness measures for word sense disambiguation based on words in context. These evaluations helped us choose the similarity and relatedness measures to use in this work. Other works, such as (Sinha and Mihalcea, 2007), use similarity or relatedness measures over graphs connecting words within a sentence. Likewise, (Navigli and Velardi, 2005) analyze the connectivity of concepts in a sentence among Structured Semantic Interconnections (SSI), graphs of relationships based on many knowledge sources. These works do not use selectors or the Web. Additionally, *target selectors* and *context selectors* provide an application for the distinction between *similarity* and *relatedness* not used in these other methods.

Several ideas distinguish this current work from our research described in (Schwartz and Gomez, 2008). The most notable aspect is that we have generalized the overall method of using Web selectors into disambiguating verbs, adverbs, and adjectives in addition to nouns. Another difference is the inclusion of selectors for adverbs. Finally, we also explore the actual impact that each type of selector has on the performance of the disambiguation algorithm.

3 Approach

In this section we describe the Web Selector algorithm such that verbs, adjectives, and adverbs are disambiguated in addition to nouns. The algorithm essentially runs in two steps: *acquisition of selectors* and *application of selectors*.

3.1 Acquisition of Selectors

Selectors are acquired for all appropriate parts of speech. Whether the selectors are used as *target selectors* or *context selectors* depends on the target word with which they are being applied. Thus, one process can be used to acquire all *noun, verb, adjective, and adverb selectors*. Additionally, noun selectors can be acquired for pronouns and proper nouns (referred to as “pro” selectors). These are regular nouns found to replace a pronoun or proper noun within their local context.

The first step in acquisition is to construct a query with a wildcard in place of the target. In our example, with ‘address’ as the target, the query is “he * the strikers at the rally.” Yahoo! Web Services¹ provides the functionality for searching the web for phrases with wildcards. Selectors are extracted from the samples returned from the web search by matching the words which take the place of the wildcard. All words not found in WordNet under the same part of speech as the target are thrown out as well as phrases longer than 4 words or those containing punctuation.

The system enters a loop where it:

- searches the web with a given query, and
- extracts selectors from the web samples.

The query is truncated and the search is repeated until a goal for the number of selectors was reached or the query becomes too short. This approach, detailed in (Schwartz and Gomez, 2008), removes select punctuation, determiners, and gradually shortens the query one word at a time. Selectors retrieved from a larger query are removed from the results of smaller queries as the smaller queries should subsume the larger query results. Some selectors retrieved for the example, with their corresponding web query are listed in Table 1.

3.2 Similarity and Relatedness

To apply selectors in disambiguation, *similarity* and *relatedness* measures are used to compare the selectors with the target word. We incorporate the use of a few previously defined measures over WordNet (Miller et al., 1993). The WordNet::Similarity package provides a flexible implementation of many of these measures (Pedersen et al., 2004). We configured WordNet::Similarity for WordNet version 2.1,

¹<http://developer.yahoo.com/search/>

*He addressed the * at the rally*

crowd:1

*He addressed * at the rally*

student:1, supporter:2

*He addressed * at the*

Council:1, Muslim:1, Saturday:1, Ugandan:1, analyst:2, attendee:20, audience:3, class:2, consumer:1, council:1, delegate:64, diplomat:2, employee:2, engineer:1, fan:1, farmer:1, globalization:1, graduate:5, guest:2, hundred:3, investor:1, issue:1, journalist:9, lawmaker:11, legislator:1, member:6, midshipman:1, mourner:1, official:2, parliamentarian:1, participant:17, patient:1, physician:18, reporter:8, sailor:1, secretary:1, soldier:3, staff:3, student:20, supporter:8, thousand:3, today:2, trader:1, troops:2, visitor:1, worker:1

*He * the strikers at the*

treat:2

*He * the strikers at*

get:1, keep:1, price:1, treat:1

Table 1: Lists of selectors for the target words ‘striker’ and ‘address’ returned by corresponding web queries.

the same version used to annotate our chosen experimental corpus.

A *relatedness* measure was used with *context selectors*, and we chose the adapted Lesk algorithm (Banerjee and Pedersen, 2002). An important characteristic of this measure is that it can handle multiple parts of speech. For *target selectors* we sought to use measures over the WordNet ontology in order to most closely measure *similarity*. An information-content (IC) measure (Resnik, 1999) was used for target selectors of nouns and verbs. However, because IC measures do not work with all parts of speech, we used the adapted Lesk algorithm as an approximation of *similarity* for adjectives and adverbs. Note that finding the best relatedness or similarity measure was outside the scope of this paper.

The following function, based on Resnik’s *word similarity* (Resnik, 1999), is used to find the max similarity or relatedness between a concept and a word (specifically between a sense of the target word, c_t and a selector, w_s).

$$\max_{sr}(c_t, w_s) = \max_{c_s \in w_s} [meas(c_t, c_s)]$$

where c_s is a sense of the selector and *meas* is a similarity or relatedness measure.

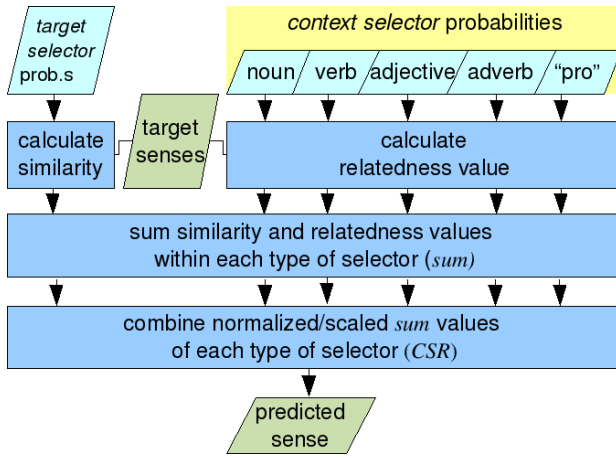


Figure 1: General flow in applying selectors to word sense disambiguation. Note that the target selectors may be any part of speech.

3.3 Application of Selectors

Next, we briefly describe the empirical basis for scoring senses of the target word. This step is outlined in Figure 1. The occurrences of selectors can be converted to a probability of a selector, w_s appearing in a web query, q :

$$p(w_s, q)$$

The senses of the target word are compared with each selector. For a given sense of the target word, c_t , the similarity or relatedness from a selector and query is computed as:

$$SR(c_t, w_s, q) = \frac{p(w_s, q) * maxsr(c_t, w_s)}{senses(w_s)}$$

where $senses(w_s)$ is the number of senses of the selector.

As the queries get shorter, the accuracy of the selectors becomes weaker. In turn, the SR value from selectors is scaled by a ratio of the web query length, wql , to the original sentence length, sl . This scaling is applied when the SR values for one target word sense are summed:

$$sum(c_t, T) = \sum_{q \in qs(T)} \sum_{w_s \in sels(q)} SR(c_t, w_s, q) * \frac{wql}{sl}$$

where $qs(T)$ represents the set of queries for a selector type, T , and w_s ranges over all selectors found with q , denoted $sels(q)$.

The general approach of disambiguation is to find the sense of a target word which is most similar to all target selectors and most related to all context selectors. This follows our assumptions about selectors given in the background section. Thus, similarity and relatedness values from different selector types (represented as $Types$) must be combined. By aggregating the normalized $sums$ from all types of selectors, we get a combined similarity/relatedness for a given target word sense:

$$CSR(c_t) = \sum_{T \in Types} scale(T) * \frac{sum(c_t, T)}{\max_{c_i \in w_t} [sum(c_i, T)]}$$

where w_t represents the set of all senses belonging to the target word, and $scale(T)$ is a coefficient used to weight each type of selector. This term is important in this work, because our experiments explore the impact of various selector types.

The top sense is then chosen by looking at the CSR of all senses. For some situations, specifically when other senses have a score within 5% of the top CSR , the difference between concepts is very small. In these cases, the concept with the lowest sense number in WordNet is chosen from among the top scoring senses.

4 Experiments

Our experiments are run over the SemEval2007 Task 7: coarse-grained English all-words. The sense inventory was created by mapping senses in WordNet 2.1 to the Oxford Dictionary of English (Navigli et al., 2007). The corpus was composed of five documents with differing domains resulting in 2269 annotated word instances. Our system runs on fine-grained WordNet senses, but evaluation is done by checking if the predicted fine-grained sense maps to the correct coarse-grained sense. Many issues associated with fine-grained annotation, such as those brought up in (Ide and Wilks, 2006) are avoided through the use of this corpus.

First, we apply the generalized Web selector algorithm in a straight-forward manner to the entire task. Then, we delve into analyzing the acquired selectors and the influence of each type of *context selector* in order to gain insights into future related work.

\mathbf{BL}_{Rand}	MED	WS	\mathbf{BL}_{MFS}
53.43	70.21	76.02	78.89

Table 2: Results as F1 Values of our system, **WS**, compared with baselines: random, \mathbf{BL}_{Rand} ; most frequent sense, \mathbf{BL}_{MFS} ; median system performance at SemEval07, **MED**.

UPV-WSD	NUS-PT	SSI
78.63	82.50	83.21

Table 3: Results as F1 Values of top performing systems for the SemEval07 Task07 (UPV = (Buscaldi and Rosso, 2007), NUS-PT = (Chan et al., 2007), and SSI = a task organizer’s system (Navigli and Velardi, 2005)).

4.1 Evaluating All Words

In this section, we seek to apply the algorithm to all instances of the testing corpus in order to compare with baselines and other disambiguation algorithms. Unless stated otherwise, all results are presented as F1 values, where $F1 = 2 * \frac{P * R}{P + R}$. For SemEval2007, all systems performed better than the random baseline of 53.43%, but only 4 of 13 systems achieved an F1 score higher than the MFS baseline of 78.89% (Navigli et al., 2007).

Table 2 lists the results of applying the generalized Web selector algorithm described in this paper in a straight-forward manner, such that all $scale(T)$ are set to 1. We see that this version of the system performs better than the median system in the SemEval07 task, but it is a little below the MFS baseline. A comparison with top systems is seen in Table 3. Our overall results were just below that of the top system not utilizing training data, (UPV-WSD (Buscaldi and Rosso, 2007)), and a little over 6 percentage points below the top supervised system (NUS-PT (Chan et al., 2007)).

The results are broken down by part of speech in Table 4. We see that adjective disambiguation was the furthest above our median point of reference, and noun disambiguation results were above the MFS baseline. On the other hand, our adverb disambiguation results appear weakest compared to the baselines. Note that we previously reported a noun sense disambiguation F1 value of 80.20% on the same corpus (Schwartz and Gomez, 2008). Current results differ because the previous work used

	N	V	A	R
MED	70.76	62.10	71.55	74.04
WS	78.52	68.36	81.21	75.48
\mathbf{BL}_{MFS}	77.44	75.30	84.25	87.50
<i>insts</i>	1108	591	362	208

Table 4: Results as F1 values (precision = recall) of our system by parts of speech (N = noun, V = verb, A = adjective, R = adverb). *insts* = disambiguation instances of each part of speech. For other keys see Table 2.

different $scale(T)$ values as well as a custom noun similarity measure.

4.2 Selector Acquisition Analysis

We examine the occurrences of acquired selectors. Listed as the column headings of Table 5, selectors are acquired for five parts of speech (*pro* is actually a combination of two parts of speech: pronoun and proper noun). The data in Table 5 is based on results from acquiring selectors for our experimental corpus. The information presented includes:

- insts* instances which the algorithm attempts to acquire selectors
- % w/ sels* percentage of instances for which selectors were acquired
- sels/inst* average number of selectors for an instance (over all *insts*)
- unique/inst* average number of unique selectors for an instance (over all *insts*)
- insts/sent* average instances in a sentence

	<i>noun</i>	<i>verb</i>	<i>adj.</i>	<i>adverb</i>	<i>pro</i>
<i>insts</i>	1108	591	362	208	370
<i>% w/ sels</i>	54.5	65.8	61.0	57.2	27.0
<i>sels/inst</i>	36.5	51.2	29.5	17.7	15.9
<i>unique/inst</i>	11.6	13.1	8.4	4.1	5.6
<i>insts/sent</i>	4.5	2.4	1.5	0.8	1.5

Table 5: Various statistics on the acquired selectors for the SemEval07 Task 7 broken down by part of speech. Row descriptions are in the text.

The selector acquisition data provides useful information. In general, *% w/ sels* was low from being unable to find text on the Web matching local context (even with truncated queries). The lowest *% w/ sels*, found for *pro*, was expected considering only nouns which replace the original words are

used (pronouns acquired were thrown out since they are not compatible with the relatedness measures). There was quite a variation in the *sels/inst* depending on the type, and all of these numbers are well below the upper-bound of 200 selectors acquired before the algorithm stops searching. It turned out that only 15.9% of the instances hit this mark. This means that most instances stopped acquiring selectors because they hit the minimum query length (5 words). In fact, the average web query to acquire at least one selector had a length of 6.7 words, and the bulk of selectors came from shorter queries (with less context from shorter queries, the selectors returned are not as strong). We refer to the combination of quantity and quality issues presented above, in general, as the *quality selector sparsity* problem.

Although quality and quantity were not ideal, when one considers data from the sentence level, things are more optimistic. The average sentence had 10.7 instances (of any part of speech listed), so when certain selector types were missing, others were present. As explained previously, the *target selector* and *context selector* distinction is made after the acquisition of selectors. Thus, each instance is used as both (exception: *pro* instances were never used as *target selectors* since they were not disambiguated). Employing this fact, more information can be discovered. For example, the average noun was disambiguated with 36.5 *target selectors*, 122.9 *verb context selectors* ($51.2 \text{ sels/inst} * 2.4 \text{ insts/sent}$), 44.3 *adjective context selectors*, 14.2 *adverb context selectors*, and 23.9 *pro context selectors*. Still, with the bulk of those selectors coming from short queries, the reliability of the selectors was not strong.

4.3 Exploring the Influence of Selector Types

This section explores the influence of each context selector on the disambiguation algorithm, by changing the value of $scale(T)$ in the previously listed *CSR* function.

Examining Table 6 reveals precision results when disambiguating instances with *target selectors*, based only on the target word’s similarity with target selectors. This serves as a bearing for interpreting results of context selector variation.

We tested how well each type of *context selector* complements the *target selectors*. Accordingly,

wsd	prec. %	insts.
N	64.08	348
V	52.86	227
A	77.36	106
R	58.39	56

Table 6: Precision when disambiguating with *target selectors* only. All instances contain target selectors and multiple senses in WordNet. (*insts.* = number of instances disambiguated.)

wsd	<i>noun</i>	<i>verb</i>	<i>adj.</i>	<i>adverb</i>	<i>pro</i>
N	272	186	120	84	108
V	211	167	110	80	103
A	97	78	50	40	34
R	47	44	30	17	26

Table 7: Instance occurrences used for disambiguation when experimenting with all types of context selectors (listed as columns). The rows represent the four parts of speech disambiguated.

$scale(target)$ was set to 1, and $scale(T)$ for all other context types were set to 0. In order to limit external influences, we did not predict words with only one sense in WordNet or instances where the *CSR* was zero (indicating no selectors). Additionally, we only tested on examples which had at least one target selector and at least one selector of the specific type being examined. This restriction ensures we are avoiding some of the *quality selector sparsity* problem described in the analysis. Nevertheless, results are expected to be a little lower than our initial tests as we are ignoring other types of selectors and not including monosemous words according to WordNet. Table 7 lists the instance occurrences for each of the four parts of speech that were disambiguated, based on these restrictions.

Figures 2 through 5 show graphs of the precision score while increasing the influence of each context selector type. Each graph corresponds to the disambiguation of a different part of speech, and each line in a graph represents one of the five types of context selectors:

1. *noun context*
2. *verb context*
3. *adjective context*
4. *adverb context*
5. *pro context*

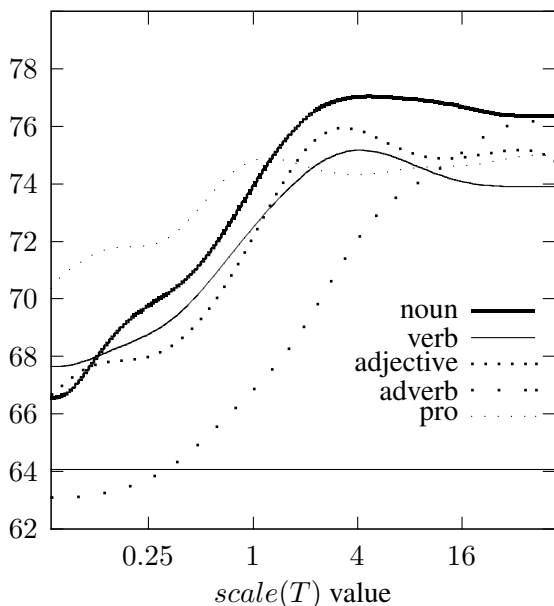


Figure 2: The **noun** sense disambiguation precision when varying the $scale(T)$ value for each type of context selector. $scale(target)$ is always 1.

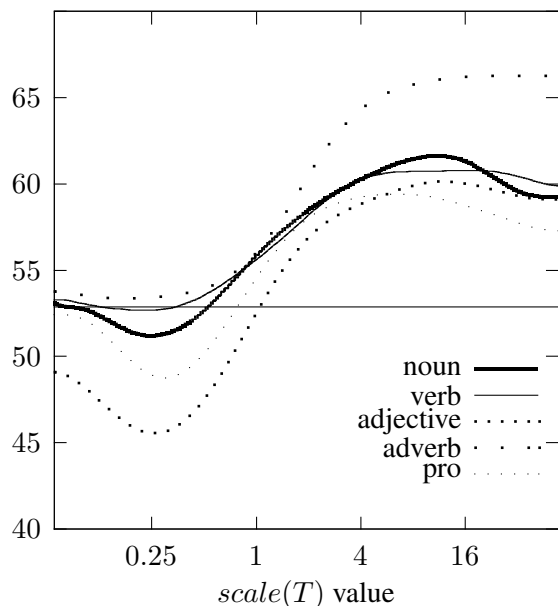


Figure 3: The **verb** sense disambiguation precision when varying the $scale(T)$ value for each type of context selector. $scale(target)$ is 1.

The lines are formed with a Bezier curve algorithm² on the precision data. The horizontal line represents the precision of only using the target selectors to disambiguate instances with target selectors. Precision either decreases or remains the same if any graph line was extended past the right-most boundary.

When examining the figures, one should note when the precision increases as the $scale$ value increases. This indicates that increases in influence of the particular type of context selector improved the results. The x-axis increases exponentially, since we would like a ratio of $scale(T)$ to $scale(target)$, and at $x = 1$ the *context selector* has the same influence as the *target selector*.

We see that all types of *context selectors* improve the results for noun and verb sense disambiguation. Thus, our inclusion of adverb context selectors was worthwhile. It is difficult to draw a similar conclusion from the adverb and adjective disambiguation graphs (Figures 4 and 5), although it still appears that the *noun context selectors* are helpful for both and the *pro context selectors* are helpful for the adjective task. We also note that most selector types

achieve highest precision above a $scale$ value of 1, indicating that the *context selector* should have more influence than the *target selectors*. This is probably due to the existence of more selectors from context than those from the target word. The results of adverb disambiguation should be taken lightly, because there were not many disambiguation instances that fit the restrictions (see Table 7).

4.4 Discussion of Future Work

Based on the results of our analysis and experiments, we list two avenues of future improvement:

1. *Automatic Alternative Query Construction*: This idea is concerned with the quality and quantity of selectors acquired for which there is currently a trade-off. As one shortens the query to receive more quantity, the quality goes down due to a less accurate local context. One may be able to side-step this trade-off by searching with alternative queries which capture just as much local context. For example, the query “He * the strikers at the rally” can be mapped into the passive transformation “the strikers were * at the rally by him”. Query

²<http://www.gnuplot.info/docs/node124.html>

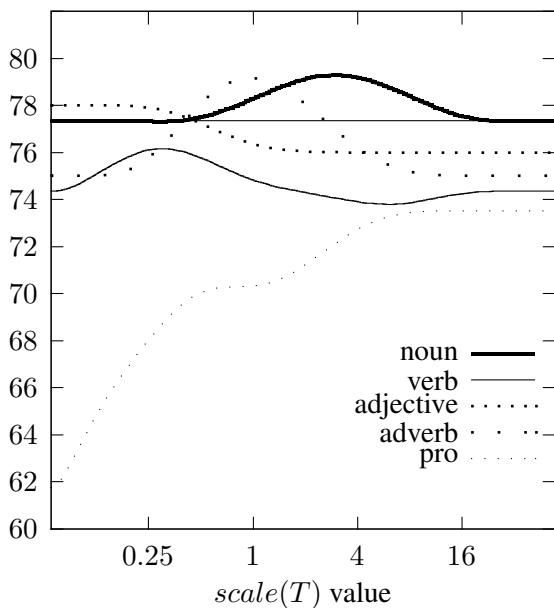


Figure 4: The **adjective** sense disambiguation precision when varying the $scale(T)$ value for each type of context selector. $scale(target)$ is 1.

reconstruction can be accomplished by using a constituent-based parser, which will help to produce syntactic alternations and other transformations such as the dative.

2. *Improving Similarity and Relatedness*: Noun sense disambiguation was the only subtask to pass the MFS baseline. One reason we suspect for this is that work in similarity and relatedness has a longer history over nouns than over other parts of speech (Budanitsky and Hirst, 2006). Additionally, the hypernym (is-a) relationship of the noun ontology of WordNet captures the notion of *similarity* more clearly than the primary relationships of other parts of speech in WordNet. Accordingly, future work should look into specific measures of *similarity* for each part of speech, and further improvement to *relatedness* measures which function across different parts of speech. A subtle piece of this type of work may find a way to effectively incorporate pronouns in the measures, allowing less selectors to be thrown out.

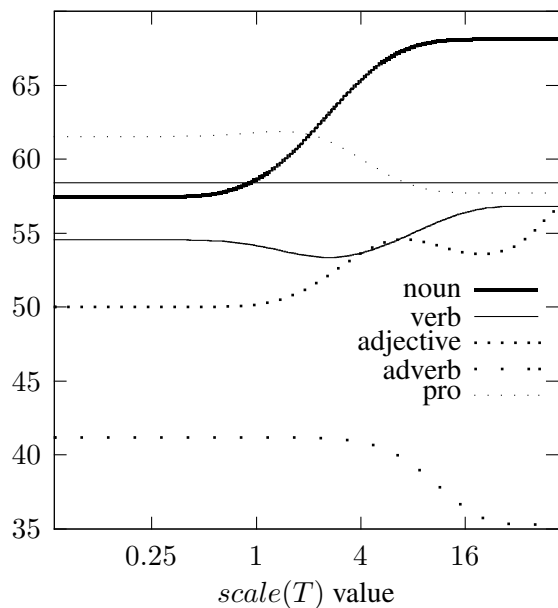


Figure 5: The **adverb** sense disambiguation precision when varying the $scale(T)$ value for each type of context selector. $scale(target)$ is 1.

5 Conclusion

We found the use of Web selectors to be a worthwhile approach to the disambiguation of other parts of speech in addition to nouns. However, results for verb, adjective, and adverb disambiguation were slightly below the most frequent sense baseline, a point which noun sense disambiguation overcomes. The use of this type of algorithm is still rich with avenues yet to be taken for improvement.

Future work may address aspects at all levels of the algorithm. To deal with a *quality selector sparsity* problem, a system might automatically form alternative web queries utilizing a syntactic parser. Research may also look into defining *similarity* measures for adjectives and adverbs, and refining the similarity measures for nouns and verbs. Nevertheless, without these promising future extensions the system still performs well, only topped by one other minimally supervised system.

6 Acknowledgement

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

References

- Eneko Agirre and David Martinez. 2004. Unsupervised WSD based on automatically retrieved examples: The importance of bias. In *Proceedings of EMNLP 2004*, pages 25–32, Barcelona, Spain, July.
- Eneko Agirre, Olatz Ansa, and David Martinez. 2001. Enriching wordnet concepts with topic signatures. In *Proceedings of the NAACL workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, Pittsburg, USA.
- Satanjeev Banerjee and Ted Pedersen. 2002. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics*. Mexico City, Mexico.
- Alexander Budanitsky and Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.
- Davide Buscaldi and Paolo Rosso. 2007. UPV-WSD : Combining different WSD methods by means of fuzzy borda voting. In *Proceedings of SemEval-2007*, pages 434–437, Prague, Czech Republic, June.
- Yee Seng Chan, Hwee Tou Ng, and Zhi Zhong. 2007. NUS-PT: Exploiting parallel texts for word sense disambiguation in the english all-words tasks. In *Proceedings of Proceedings of SemEval-2007*, pages 253–256, Prague, Czech Republic, June.
- Nancy Ide and Yorick Wilks, 2006. *Word Sense Disambiguation: Algorithms And Applications*, chapter 3: Making Sense About Sense. Springer.
- Dekang Lin. 1997. Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the 35th annual meeting on Association for Computational Linguistics*, pages 64–71.
- David Martinez, Eneko Agirre, and Xinglong Wang. 2006. Word relatives in context for word sense disambiguation. In *Proceedings of the 2006 Australasian Language Technology Workshop*, pages 42–50.
- Rada Mihalcea and Dan I. Moldovan. 1999. An automatic method for generating sense tagged corpora. In *Proceedings of AAAI-99*, pages 461–466.
- Rada Mihalcea. 2002. Bootstrapping large sense tagged corpora. In *Proceedings of the 3rd International Conference on Languages Resources and Evaluations LREC 2002*, Las Palmas, Spain, May.
- George Miller, R. Beckwith, Christiane Fellbaum, D. Gross, and K. Miller. 1993. Five papers on wordnet. Technical report, Princeton University.
- Roberto Navigli and Paola Velardi. 2005. Structural semantic interconnections: A knowledge-based approach to word sense disambiguation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(7):1075–1086.
- Roberto Navigli, Kenneth C. Litkowski, and Orin Har- graves. 2007. Semeval-2007 task 07: Coarse-grained english all-words task. In *Proceedings of SemEval-2007*, pages 30–35, Prague, Czech Republic. Association for Computational Linguistics.
- Siddharth Patwardhan, S. Banerjee, and T. Pedersen. 2003. Using Measures of Semantic Relatedness for Word Sense Disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pages 241–257, Mexico City, Mexico, February.
- Ted Pedersen, S. Patwardhan, and J. Michelizzi. 2004. WordNet::Similarity - Measuring the Relatedness of Concepts. In *Human Language Technology Conference of the NAACL Demonstrations*, pages 38–41, Boston, MA, May.
- R. Rada, H. Mili, E. Bicknell, and M. Blettner. 1989. Development and application of a metric on semantic nets. In *IEEE Transactions on Systems, Man and Cybernetics*, volume 19, pages 17–30.
- 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.
- Hansen A. Schwartz and Fernando Gomez. 2008. Acquiring knowledge from the web to be used as selectors for noun sense disambiguation. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 105–112, Manchester, England, August.
- Ravi Sinha and Rada Mihalcea. 2007. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. Irvine, CA, September.
- Deniz Yuret. 2007. KU: Word sense disambiguation by substitution. In *Proceedings of SemEval-2007*, pages 207–214, Prague, Czech Republic, June.