# Improving Supervised Sense Disambiguation with Web-Scale Selectors

*H. Andrew Schwartz*[1] *Fernando Gomez*[2] *Lyle H. Ungar*[1]
(1) University of Pennsylvania, Philadelphia, PA USA
(2) University of Central Florida, Orlando, FL USA
hansens@seas.upenn.edu, gomez@eecs.ucf.edu, ungar@cis.upenn.edu

ABSTRACT
This paper introduces a method to improve supervised word sense disambiguation performance by including a new class of features which leverage contextual information from large unannotated corpora. This new feature class, *selectors*, contains words that appear in other corpora with the same local context as a given lexical instance. We show that support vector sense classifiers trained with selectors achieve higher accuracy than those trained only with standard features, producing error reductions of 15.4% and 6.9% on standard coarse-grained and fine-grained disambiguation tasks respectively. Furthermore, we find an error reduction of 9.3% when including selectors for the classification step of named-entity recognition over a representative sample of OntoNotes. These significant improvements come free of any human annotation cost, only requiring unlabeled Web-Scale corpora.

KEYWORDS: word sense disambiguation, lexical semantics, semi-supervised learning.

*Proceedings of COLING 2012: Technical Papers*, pages 2423–2440,
COLING 2012, Mumbai, December 2012.

## 1 Introduction

Supervised word sense disambiguation (*WSD*) systems often rely directly on the local contexts in which target words appear. For example, the state-of-the-art system of Zhong et al. (2008) uses features based on collocations centered on the target word. Models relying on such features do well with copious amounts of training data, but they are prone to errors when the local context of a test instance differs from local context observed during training. Consider the sentences below.

1. *The workers loaded the **port** onto the ship this morning.*
2. *She purchased a couple of bottles of **port** from the store.*
3. *The couple enjoyed their richly-flavored **port**.*

Though referring to the same sense of 'port', "a sweet dark-red dessert wine" (Miller et al., 1993), it is difficult to connect any two instances based on local context; the parts-of-speech even differ substantially. Models for disambiguation can benefit from the addition of a feature that does not rely directly on the local context.

We present a new class of features which encodes an abstraction of a word's context, rather than encoding contents of the local context itself. We refer to this new feature class as *selectors*, borrowing the term from an approach to knowledge-based (unsupervised) word sense disambiguation which uses the idea of searching for words that share the same context (Lin, 1997; Schwartz and Gomez, 2008). More precisely, selectors are words that show up in the same local context as a given instance of another word. For example, selectors for 'port' in sentence 1 might be 'bottles', 'crates', 'passengers', 'wine', 'luggage', etc. Considering that the other sentences may share some of the same selectors such as 'bottles' or 'wine', one can see how this abstraction of context to selectors can be beneficial. Figure 1 demonstrates mapping the context from one instance to selectors, which match the selectors of another instance. In this sense, it is the contexts (or word instances) that have selectors rather than the words themselves.

The contribution of this paper is the introduction of a novel and effective type of feature that improves *WSD* accuracy at no cost in human annotation. Rather than requiring more examples of labeled context to match a given test instance, we need only to match against an orders-of-magnitude-larger unlabeled set of data. Because *selectors* leverage unlabeled data, their inclusion in a supervised system constitutes semi-supervised learning.

The paper proceeds with a discussion of related work in semi-supervised *WSD* and the use of web-scale data in language processing (Section 2). Then, we present our approach to acquiring selectors as features from n-grams, and show how we translate selectors into features (Section 3). The effectiveness of selectors is evaluated within supervised word sense disambiguation classifiers over the SemEval-2007 Task 17 (Pradhan et al., 2007), Senseval 3 English Lexical Sample (Mihalcea et al., 2004), and OntoNotes 4 (Weischedel et al., 2011) (Section 4). We also test selectors as features for the classification step of named-entity recognition over a representative sample of OntoNotes. Lastly, we discuss the robustness of selectors as features by inspecting actual instances from our experimental corpus (Section 5).

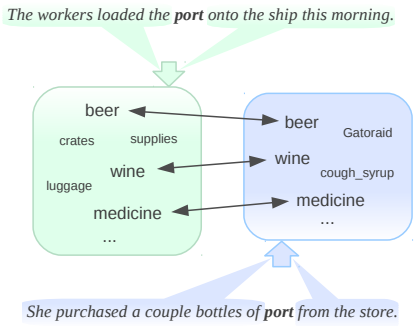


Figure 1: Word instances which do not share context can share selectors.

## 2 Related Work

The idea of improving a supervised classifier by utilizing unlabeled data has been investigated at different levels. For example, other approaches to disambiguation have used bootstrapped samples (Yarowsky, 1995; Mihalcea and Moldovan, 1999; Mihalcea, 2004; Pham et al., 2005), Wikipedia concepts (Mihalcea, 2007), or parallel corpora (Chan et al., 2007). Most of these approaches, which are considered semi-supervised learning (Zhu, 2008), exploit some facet of unannotated text to collect more training instances. Rather than produce more training instances, we introduce a method to leverage massive unlabeled corpora to create a richer and robust set of features.

One can contrast *selectors* with clusters of words formed via context or distributional similarity (for seminal examples see (Brown et al., 1992; Pereira et al., 1993; Lin, 1998; Schütze, 1998; Pantel and Lin, 2002)). Distributional clusters are made up of words that appear in similar contexts to each other, whereas selectors are words which show up in the specific context of a single instance. In other words, selectors are instance-specific while distributional clusters are created based on observing many instances of context. This key difference should become more clear when we present our method of acquiring selectors.

The traditional use of selectors is in knowledge-based word sense disambiguation systems, not utilizing training data. In Lin (1997), dependency relationships over a small corpus were used to find noun selectors. We previously extended this to the Web, treating context as surrounding text and introduced the ideas of acquiring selectors for additional parts-of-speech as well as for words in context in addition to the target word (Schwartz and Gomez, 2008, 2009). Similar to selectional preferences (Resnik, 1997), *selectors* essentially indicate the types of concepts expected in a given syntactic or grammatical position. In these knowledge-based approaches, disambiguation is performed by computing the semantic distance between *selectors* and senses of the target word. These approaches rely on both a knowledge source such as WordNet (Miller et al., 1993) and a semantic distance metric. In contrast, in the current approach we do not need such a knowledge source or similarity judgments, and since our approach is data-driven, selectors function as an abstraction of word instance context rather than as a list of semantically similar words. Our current goal is to get the most out of supervised training data by leveraging unannotated data via selectors (no use of a knowledge-base or similarity metrics). Consequently,

our system achieves state-of-the-art results in line with top supervised systems while our earlier knowledge-based approaches produce results in line with systems not utilizing training data.

A couple previous works have integrated unannotated data as features into supervised disambiguation systems. Dligach and Palmer (2008) used *dynamic dependency neighbors*, a feature encoding verbs with the same object, according to a dependency parsed corpus, as a given target verb in a verb WSD task. Besides our method not being limited to verbs, selectors are much more specific than dependency neighbors; They are found by matching a larger context and from a much larger, web-scale, dataset. Cárcamo et al. (2008) adapt the predominant sense method of McCarthy et al. (2004) to find the best sense choice for a word instance rather than it's most common sense over a corpus. Yuret (2007) leveraged web-scale data to acquire probability distributions of *substitutes* being within the same context as target instances. Unlike selectors which are open-ended, substitutes were chosen from an *a priori* word list derived from thesauri, and contextual part-of-speech was not considered. Additionally, the substitutes' probability distribution itself was the entire feature set, rather than used to supplement an existing feature set, and the resulting accuracies were lower than those we find with selectors.

Our approach utilizes web-scale N-grams, a source of unlabeled data which has previously been used for many other supervised lexico-semantic tasks including delimiting named entities, preposition selection, spelling correction, search query processing, adjective ordering, verb POS disambiguation, and noun compound bracketing (Downey et al., 2007; Bergsma et al., 2009; Huang et al., 2010; Bergsma et al., 2010). All of these systems utilized n-grams to find frequency information for specific n-grams. In contrast, we use the n-grams as a source for acquiring sets of lexical data (selectors), where we search with context and ask for the missing piece rather than search for a complete n-grams. We believe this is the first work to use web-scale N-grams as a source for selectors; motivation for using this source is discussed in the next section.

## 3 Acquiring Selectors

A selector is a word which appears in the same local context as a given instance of a *focus word*. For example, in the sentence below, with 'port' as the *focus word*, one might find selectors such as 'bottles', 'cargo', 'crates', 'wine', 'passengers', or 'supplies'.

> *The workers loaded the* ***port*** *onto the ship last night.*

More formally, for a given word instance, $w_i$, selectors are found based on the particular context of $w_i$. What defines the context may vary from syntactic or dependency relations (i.e., other nouns which are objects of the verb 'loaded') to simple sequences of tokens (e.g.,finding words that fill in the blank in "The workers loaded the ___ onto the ship last night.").

### 3.1 Approach

We find selectors by searching for sequences of tokens in the Google N-grams version 2, which contains 4.1 billion n-grams that were automatically part-of-speech tagged (Lin et al., 2010). The primary reason we chose web-scale N-grams as a source is because it has become difficult to get selectors via search engines.[1] Still, using web-scale n-grams for context searches has advantages: there is a decent likelihood of finding selectors for a given instance, the search

[1]The Web search engines which support wildcard queries no longer run public APIs or allow scripted access.

*Workers loaded the **port** onto the ship last night.*
workers loaded $\langle det \rangle$? ($\langle noun \rangle$+) onto
loaded $\langle det \rangle$? ($\langle noun \rangle$+) onto
($\langle noun \rangle$+) onto $\langle det \rangle$ ship last

---

*My objective was to **fight** as a mother for what I hold dearest.*
was to ($\langle verb \rangle$+) as $\langle det \rangle$
to ($\langle verb \rangle$+) as $\langle det \rangle$? mother
objective was to ($\langle verb \rangle$+)

---

*The new economy in the US depends heavily, for one thing, on a deep foundation of basic scientific research, which comes up with revolutionary **products** like genetically modified foods.*
with revolutionary ($\langle noun \rangle$+) like genetically
revolutionary ($\langle noun \rangle$+) like
($\langle noun \rangle$+) like genetically modified

---

*...which comes up with **revolutionary** products like...*
up with ($\langle adj \rangle$+) products like
up with ($\langle adj \rangle$+) products
with ($\langle adj \rangle$+) products

Table 1: Example search sequences produced for the given focus word focus word (***in bold***) and context (*in italic*).'()' surrounds the *focus word*, '?' implies optional match and '+' allows multiple matches. The bottom three examples are from our experimental corpus.

process is offline, this version of the Google N-grams provides part-of-speech information, and they have been shown helpful for other lexico-semantic tasks (Bergsma et al., 2009; Huang et al., 2010; Bergsma et al., 2010). On the downside, because the Google N-grams are at most 5-grams, the selectors can only be found using a relatively small context. – up to 4 tokens. For this first investigation of selectors as features we think this trade-off is worthwhile.

We search the n-grams by constructing 3 to 5 token sequences consisting of words or part-of-speech (*POS*) tags. Determiners, conjunctions, possessives, and symbols in the sequence are replaced with their *POS* tag, and determiners are also marked as optional if they do not begin or end the phrase. The slot of the *focus word*, the word for which selectors are being acquired, is restricted by *POS* and permitted to match multi-word phrases (taking the head-word as the selector in such cases). Examples of search sequences are given in Table 1.

Searching based on all sequences can be expensive in terms of disk IO, so the sequences are sorted such that the process can be halted once enough selectors have been found. In particular, we define four *criteria* of informative value for a given sequence $seq$:

1. the number of tokens in $seq$:
$length(seq) = \frac{|tokens(seq)|}{max\_tokens}$
2. the number of content words (noun, verbs, adjective, or adverbs):
$nvar(seq) = \frac{|nouns(seq)|+|verbs(seq)|+|adjectives(seq)|+|adverbs(seq)|}{max\_tokens}$

3. the distance from the focus word to the center: $center(seq) = 1 - \frac{|before(seq) - after(seq)|}{|tokens(seq)|}$

4. if the *focus word* is an edge of $seq$:
$$\neg edge(seq) = \begin{cases} 0, & \text{if focus word is at front or back} \\ 1, & \text{otherwise} \end{cases}$$

where $max\_tokens = 5$, the maximum number of tokens in a sequence, and $before$ / $after$ are the number of token before and after the focus word. The overall informative value is defined as the sum of weighted ($\alpha_i$) *criteria* ($c_{1..4} = [length(seq), nvar(seq), center(seq), \neg edge(seq)]$ ):

$$info(seq) = \sum_{i=1}^{4} \alpha_i c_i$$

Next, we iterate through the list of sorted search sequences in order to aggregate selector($s$) frequencies. During aggregation, the selector frequencies are normalized and weighted by $info(seq)$:

$$score(s) = \sum_{seq \in seqs} info(seq) * \frac{freq(s)}{\max_{s' \in sels(seq)} freq(s')}$$

where $sels(seq)$ is the set of selectors found when searching with sequence $seq$. This favors both selectors occurring with multiple sequences as well as those found based on a more informative context. In practice we break the aggregation loop one iteration after acquiring a soft minimum ($k$) of selectors to improve runtime. [2]

## 3.2 Selectors as Features.

We have described acquisition of selectors for an arbitrary *focus word* instance. In order to use selectors as features, we acquire selectors for all target words (words being disambiguated) and encode the top $k$, according to $score(s)$, as binary features. We selected $k = 50$ as well as weightings $\alpha_{1..4} = [0.2, 0.2, 0.1, 0.5]$ after cross-validating over the *training set* (defined in Section 4).

# 4 Experiments

We evaluate whether the *selector* class of features can benefit *WSD* classifiers above and beyond a standard set of features in a variety of datasets and situations. Supervised classifiers are trained with and without utilization of *selectors* and we record a simple accuracy of $\frac{|correct\_instances|}{|all\_instances|} * 100$ of the testing data.[3] In particular, we use support vector classifiers implemented with Scikit-learn (Pedregosa et al., 2011) with a radial basis kernel and other parameters set via 5-fold cross-validation over the training set. As a standard point of comparison, *most frequent sense* ($MFS$) accuracy is also reported, indicating the testing accuracy if the system always predicted the most common sense according to the training data. As often noted, state-of-the-art supervised systems often perform just above the $MFS$ (Navigli et al., 2007; Pradhan et al., 2007).

[2] An implementation of this method is included in supplementary data.

[3] $accuracy = precision = recall$ under the standard (SemEval) definition of precision and recall for $WSD$, and because we attempt all instances of our samples.

## 4.1 Data Sets

We test *selectors* over three sense-annotated corpora. For our primary corpus, we use the SemEval-2007 Task 17: Lexical Sample (Pradhan et al., 2007) (results in sections 4.3.1 and 4.3.3). This corpus is an early selection from the Wall Street Journal portion of OntoNotes (Weischedel et al., 2011), and contains coarse-grained noun and verb senses. We also experiment over the Senseval-3 English Lexical Sample data(Mihalcea et al., 2004), containing fined-grained noun, verb, and adjective sense annotations over selections of the British National Corpus (Clear, 1993) (section 4.3.2). The inclusion of adjectives, fine-grained senses, and difference in corpus gives us a more robust evaluation of selectors. Lastly, we experiment with random samples over portions of the full Ontonotes 4.0 in order to test on out-of-domain data and to examine if selectors help for another lexico-semantic task: named-entity classification. Details about the OntoNotes test sets are included when discussing those results (sections 4.3.4 and 4.3.5).

## 4.2 Baseline Features

As a consistent baseline throughout our experiments, we use the same features as Zhong et al. (2008)'s state-of-the-art system, first explored by Lee and Ng (2002). These features give the best published results that we are aware of over the Wall Street Journal portion of OntoNotes, plus they are the common denominator in many high-performance supervised WSD systems (Cai et al., 2007; Chan et al., 2007; Zhong et al., 2008).

- **collocations** (*coll*). Tokens relative to the target, denoted $c_{i,j}$, starting at $i$; ending at $j$.
  1-grams: $c_{-1,-1}$, $c_{+1,+1}$, $c_{-2,-2}$, $c_{+2,+2}$,
  2-grams: $c_{-2,-1}$, $c_{+1,+2}$,
  3-grams: $c_{-3,-1}$, $c_{+1,+3}$, $c_{-1,+1}$,
  4-grams: $c_{-2,+1}$, $c_{-1,+2}$
- **parts-of-speech** (*pos*). The part-of-speech for the following words relative to the target word: $p_{-3}, p_{-2}, p_{-1}, p_0, p_{+1}, p_{+2}, p_{+3}$ (0 is the target word).
- **surrounding words** (*surr*). The bag-of-words from the current, previous, and next sentence.

## 4.3 Results

### 4.3.1 SemEval-2007

Table 2 shows the results with and without selectors over the SemEval-2007 corpus. We see that including selectors improves performance over a state-of-the-art set of features with a significant ($p < 0.01$) error reduction of 15.4%.[4] This puts our system just behind the top system participating in SemEval-2007, NUS-ML (Cai et al., 2007), which achieved an accuracy of 88.7. Moreover, we see improvements from selectors for both nouns and verbs.

Tables 3 and 4 break the results down for each word. Though it is possible for selectors to introduce noise leading to occasional errors, we see that both words with many training instances as well as those with fewer ones can benefit from selectors. We will inspect a couple instances where selectors helped prediction in Section 5.

[4] **error reduction**= $\frac{(1-acc1)-(1-acc2)}{1-acc1}$, where $acc1$ and $acc2$ represent the two accuracies.

| | **base** | **w/ sels** | ***mfs*** | ***tests*** |
|---|---|---|---|---|
| noun | 87.9 | **91.7** | 80.9 | *2559* |
| verb | 83.3 | **83.7** | 76.5 | *2292* |
| both | 85.7 | **87.9** | 78.8 | *4851* |

Table 2: Classifier accuracies without (**base**) and with the selector class of features (**w/ sels**) over SemEval-2007 Task 17. (*mfs*: accuracy of classifying with the most frequent sense of the training data, *tests*: number of instances in the test set.)

| word | **base** | **w/ sels** | ***mfs*** | ***tests*** | ***trains*** |
|---|---|---|---|---|---|
| area-n | 78.4 | 83.8 | 70.3 | 37 | 326 |
| authority-n | 81.0 | 81.0 | 23.8 | 21 | 90 |
| base-n | 40.0 | **70.0** | 10.0 | 20 | 92 |
| bill-n | 98.0 | 98.0 | 75.5 | 102 | 404 |
| capital-n | 96.5 | 96.5 | 96.5 | 57 | 278 |
| carrier-n | 71.4 | 71.4 | 71.4 | 21 | 111 |
| chance-n | **73.3** | 60.0 | 40.0 | 15 | 91 |
| condition-n | 79.4 | 79.4 | 76.5 | 34 | 132 |
| defense-n | 42.9 | **61.9** | 28.6 | 21 | 120 |
| development-n | 65.5 | **79.3** | 62.1 | 29 | 180 |
| drug-n | 89.1 | **91.3** | 87.0 | 46 | 205 |
| effect-n | 86.7 | **93.3** | 76.7 | 30 | 178 |
| exchange-n | 86.9 | 86.9 | 73.8 | 61 | 363 |
| future-n | **95.2** | 94.5 | 86.3 | 146 | 350 |
| hour-n | 89.6 | **91.7** | 89.6 | 48 | 187 |
| job-n | **82.1** | 79.5 | 82.1 | 39 | 188 |
| management-n | 88.9 | **93.3** | 71.1 | 45 | 284 |
| move-n | 97.9 | 97.9 | 97.9 | 47 | 270 |
| network-n | 96.4 | **98.2** | 90.9 | 55 | 152 |
| order-n | 91.2 | 91.2 | 91.2 | 57 | 346 |
| part-n | **91.5** | 90.1 | 66.2 | 71 | 481 |
| people-n | 90.4 | **93.9** | 90.4 | 115 | 754 |
| plant-n | 98.4 | 98.4 | 98.4 | 64 | 347 |
| point-n | 90.7 | **93.3** | 81.3 | 150 | 469 |
| policy-n | 97.4 | 97.4 | 97.4 | 39 | 331 |
| position-n | 68.9 | **88.9** | 46.7 | 45 | 268 |
| power-n | 85.1 | **89.4** | 27.7 | 47 | 251 |
| president-n | **98.7** | 98.3 | 72.9 | 177 | 879 |
| rate-n | 88.3 | **90.3** | 86.2 | 145 | 1009 |
| share-n | 97.1 | **97.7** | 97.1 | 525 | 2534 |
| source-n | 80.0 | **88.6** | 37.1 | 35 | 152 |
| space-n | 92.9 | **100.0** | 78.6 | 14 | 67 |
| state-n | 79.2 | **80.6** | 79.2 | 72 | 617 |
| system-n | 68.6 | **72.9** | 48.6 | 70 | 450 |
| value-n | 98.3 | 98.3 | 98.3 | 59 | 335 |

Table 3: Classifier accuracies for each noun of the SemEval-2007 Task 17 test set. *trains*: number of training instances.

| word | **base** | **w/ sels** | ***mfs*** | ***tests*** | ***trains*** |
|---|---|---|---|---|---|
| affect-v | 100.0 | 100.0 | 100.0 | 19 | 45 |
| allow-v | **97.1** | 91.4 | 97.1 | 35 | 108 |
| announce-v | 100.0 | 100.0 | 100.0 | 20 | 88 |
| approve-v | **91.7** | 83.3 | 91.7 | 12 | 53 |
| ask-v | 74.1 | **87.9** | 51.7 | 58 | 348 |
| attempt-v | 100.0 | 100.0 | 100.0 | 10 | 40 |
| avoid-v | 100.0 | 100.0 | 100.0 | 16 | 55 |
| begin-v | 66.7 | **72.9** | 56.2 | 48 | 114 |
| believe-v | 80.0 | **83.6** | 78.2 | 55 | 202 |
| build-v | 73.9 | **78.3** | 73.9 | 46 | 119 |
| buy-v | **80.4** | 78.3 | 76.1 | 46 | 164 |
| care-v | 42.9 | 42.9 | 28.6 | 7 | 69 |
| cause-v | 100.0 | 100.0 | 100.0 | 47 | 73 |
| claim-v | 80.0 | 80.0 | 80.0 | 15 | 54 |
| come-v | 32.6 | **51.2** | 23.3 | 43 | 186 |
| complain-v | 85.7 | 85.7 | 85.7 | 14 | 32 |
| complete-v | 93.8 | 93.8 | 93.8 | 16 | 42 |
| contribute-v | **83.3** | 72.2 | 50.0 | 18 | 35 |
| describe-v | 100.0 | 100.0 | 100.0 | 19 | 57 |
| disclose-v | 92.9 | 92.9 | 92.9 | 14 | 55 |
| do-v | 90.2 | **93.4** | 90.2 | 61 | 207 |
| end-v | 66.7 | **90.5** | 52.4 | 21 | 135 |
| enjoy-v | 57.1 | 42.9 | 57.1 | 14 | 56 |
| estimate-v | 100.0 | 100.0 | 100.0 | 16 | 74 |
| examine-v | 100.0 | 100.0 | 100.0 | 3 | 26 |
| exist-v | 100.0 | 100.0 | 100.0 | 22 | 52 |
| explain-v | 88.9 | 88.9 | 88.9 | 18 | 85 |
| express-v | 100.0 | 100.0 | 100.0 | 10 | 47 |
| feel-v | 68.6 | **72.5** | 68.6 | 51 | 347 |
| find-v | 82.1 | **85.7** | 82.1 | 28 | 174 |
| fix-v | 50.0 | 50.0 | 50.0 | 2 | 32 |
| go-v | **70.5** | 63.9 | 45.9 | 61 | 244 |
| grant-v | 80.0 | 80.0 | 80.0 | 5 | 19 |
| hold-v | 50.0 | **54.2** | 37.5 | 24 | 129 |
| hope-v | 100.0 | 100.0 | 100.0 | 33 | 103 |
| improve-v | 100.0 | 100.0 | 100.0 | 16 | 31 |
| join-v | 38.9 | 38.9 | 38.9 | 18 | 68 |
| keep-v | 56.2 | **58.8** | 56.2 | 80 | 260 |
| kill-v | 87.5 | 87.5 | 87.5 | 16 | 111 |
| lead-v | **69.2** | 66.7 | 38.5 | 39 | 165 |
| maintain-v | 90.0 | **100.0** | 90.0 | 10 | 61 |
| need-v | 91.1 | 91.1 | 71.4 | 56 | 195 |
| negotiate-v | 100.0 | 100.0 | 100.0 | 9 | 25 |
| occur-v | 90.9 | **95.5** | 86.4 | 22 | 47 |
| prepare-v | **94.4** | 88.9 | 77.8 | 18 | 54 |
| produce-v | 75.0 | 75.0 | 75.0 | 44 | 115 |
| promise-v | 75.0 | **100** | 75.0 | 8 | 50 |
| propose-v | 85.7 | **92.9** | 85.7 | 14 | 34 |
| prove-v | 54.5 | **81.8** | 68.2 | 22 | 49 |
| purchase-v | 100 | 100.0 | 100.0 | 15 | 35 |
| raise-v | 29.4 | **50.0** | 14.7 | 34 | 147 |
| recall-v | 86.7 | 86.7 | 86.7 | 15 | 49 |
| receive-v | 95.8 | 95.8 | 95.8 | 48 | 136 |
| regard-v | 78.6 | 78.6 | 71.4 | 14 | 40 |
| remember-v | 100.0 | 100.0 | 100.0 | 13 | 121 |
| remove-v | 100.0 | 100.0 | 100.0 | 17 | 47 |
| replace-v | 100.0 | 100.0 | 100.0 | 15 | 46 |
| report-v | 91.4 | **94.3** | 91.4 | 35 | 128 |
| rush-v | 100.0 | 100.0 | 100.0 | 7 | 28 |
| say-v | 98.7 | 98.7 | 98.7 | 541 | 2161 |
| see-v | 44.4 | **59.3** | 44.4 | 54 | 158 |
| set-v | 47.6 | **59.5** | 28.6 | 42 | 174 |
| start-v | 44.7 | **52.6** | 44.7 | 38 | 214 |
| turn-v | 51.6 | **58.1** | 38.7 | 62 | 340 |
| work-v | 60.5 | **67.4** | 55.8 | 43 | 230 |

Table 4: Classifier accuracies for each verb of the SemEval-2007 Task 17 test set.

| | **base** | **w/ sels** | ***mfs*** | ***tests*** |
|---|---|---|---|---|
| noun | 68.5 | **72.1** | 54.1 | *1766* |
| verb | 72.0 | 72.4 | 57.9 | *1927* |
| adjective | 49.4 | **53.4** | 54.7 | *148* |
| all | 69.4 | **71.5** | 56.1 | *3841* |

Table 5: Sense classifier accuracies without (**base**) and with the selector class of features (**w/ sels**) over Seneval-3: English Lexical Sample. (*mfs*: accuracy of classifying with the most frequent sense of the training data, *tests*: number of instances in the test set.)

#### 4.3.2 Senseval-3

We also tested selectors as features over the Senseval-3 data(Mihalcea et al., 2004) to get a more robust idea of their impact. The instances in this sample come from a difference corpus, the British National Corpus, include fine-grained sense annotations, and a limited number of adjectives.

Examining the results in Table 5, we see an improvement from using selectors over the baseline for all three parts-of-speech. Overall error reductions is 6.9%. Selectors seem to help the most for both nouns and adjectives, but in the case of adjectives we actually see the *mfs* just outperforms the supervised systems. We suspect this is partly due to the average adjective having many more possible senses (10.2, versus 5.8 for nouns and 6.3 for verbs), and one should also keep in mind the small number of adjective examples.

#### 4.3.3 Feature Impact Analysis

Results discussed thus far imply *selectors* are contributing information beyond that of the standard set of features. However, since selectors represent an abstraction of context and the baseline features encode various contextual information, it is possible that all information from certain baseline features is subsumed by selectors. In this experiment, we try to understand the type of information being contributed by selectors by observing accuracies when features are removed.

Table 6 shows accuracy results when building classifiers with all combinations of feature types. For these tests, we used the SemEval-2007 data set, the larger and more recent of the two previously mentioned evaluation data sets. We see a clear benefit from the inclusion of selectors across the board. Interestingly, we see that although *surr* class of features gets the system beyond the *mfs* baseline, it seems to provide more distractions than help once other features are included as well. In fact, our best results come from the combination of collocations, part-of-speech information, and selectors with an accuracy of 88.1.

#### 4.3.4 Out-of-Domain Test Data

It is often noted that *WSD* systems perform poorly on test data from a different domain than that of the training data (Zhong et al., 2008; Agirre et al., 2010). We examine whether selectors keep their benefit when tested on out-of-domain data over a portion of OntoNotes 4.0 (Weischedel et al., 2011). We put together all occurrences of a random selection of 100 nouns and verbs over three portions of OntoNotes: The Wall Street Journal ***(WSJ)***, Xinhua New Agency ***(Xh)***, and Sinorama Magazine ***(Sr)***. The Xinhua and Sinorama corpora correspond to a different *source* of newswire data and a different *genre* (magazine) respectively. As is standard, we used

| feature types | accuracy w/o sels | w/ sels | err reduc |
|---|---|---|---|
| *coll* | 86.3 | **87.9** | 11.7 % |
| *pos* | 83.8 | **86.3** | 15.3 % |
| *surr* | 82.5 | **86.8** | 24.6 % |
| *coll, pos* | 86.9 | **88.2** | 9.9 % |
| *pos, surr* | 86.0 | **87.7** | 12.1 % |
| *surr, coll* | 85.5 | **87.4** | 13.1 % |
| *coll, pos, surr* | 85.7 | **87.9** | 15.4 % |
| *sels* alone | - | 84.7 | - |
| (***mfs***) | 78.8 | - | - |
| mean err reduc | - | - | 14.5 % |

Table 6: Accuracy of classifier utilizing all combinations of feature types on the SemEval-2007 Task 17 test set. **err reduc** is the error reduction when using selectors. Refer to section 4.2 for feature type identifiers.

| | **base** | **w/ sels** | ***mfs*** | ***tests*** |
|---|---|---|---|---|
| $WSJ$ | 82.5 | **84.3** | 80.7 | *166* |
| $Xh$ | 77.1 | **78.2** | 75.4 | *564* |
| $Sr$ | 58.1 | 58.8 | 46.8 | *816* |

Table 7: Accuracy of the classifiers when training on the $WSJ$, and applying to another source of news ($Xh$) or another genre of text ($Sr$) within OntoNotes 4.

samples from sections 02-21 of *WSJ* as training, while samples from section 22 of *WSJ* plus all sections of *Xh* and *Sr* were used for testing. We decided to use OntoNotes because our main testing corpus, SemEval-2007 Task 17, is itself derived from OntoNotes, though it lacked multiple genres of text.

We see from Table 7 that *selectors* still give an improvement in the case of another *source* of newswire. When moving to a more distant domain, such as another *genre*, the improvement still exists though it is no longer significant. The difficulty of the out-of-domain task is exemplified by lower $MFS$ values, which are still based on the most frequent sense in the training data (always $WSJ$ in this case). The results demonstrate relative robustness across minor shifts in domain, and potential for greater success if one combined them with a domain-adaptation technique.

#### 4.3.5 Named Entity Classification

We believe *selectors* can benefit other supervised lexical disambiguation tasks. In this experiment, we seek preliminary evidence for such a belief based on improving the classification step of named entity recognition.

In *named entity classification*, one is given a noun phrase representing an entity with its context, and one attempts to classify the named entity into a variety of classes. We build a classifier which labels entities with one of the 18 classes provided by OntoNotes. We sample 1000 randomly selected sentences from The Wall Street Journal, Xinhua, and Sinorama portions of OntoNotes. The data is divided into training and testing samples:

| base | w/ sels | *mfc* | *tests* |
|---|---|---|---|
| 85.0 | **86.4** | 20.2 | 259 |

Table 8: Named-entity classifier accuracies without (**base**) and with the selector class of features (**w/ sels**) across a random sample of the $WSJ$, $Xh$ and $Sr$ portions of OnotNotes. (*mfc*: accuracy of predicting the most frequent named entity class in training data, *tests*: number of instances in the test set.)

- The Wall Street Journal ***(WSJ)***: *sections 02 - 21(train); section 22(test)*
- Xinhua New Agency ***(Xh)***: *sections 0000 - 0209(train); sections 0210 - 0325(test)*
- Sinorama Magazine ***(Sr)***: *sections 1001 - 1059(train); sections 1060 - 1078(test)*

For the $WSJ$ we stick with standard training and test sets, while we divide $Xh$ and $Sr$ corpora similarly. Out of the 1,000 randomly selected sentences across these corpora there are 2,106 total named entity instances: 1,847 training examples and 259 test examples. We find this to be a representative sample of the $WSJ$, $Xh$, and $Sr$ portions of OntoNotes [5].

We choose our features by looking at the most common types of features used during the CoNLL-2003 Shared Task in Named Entity Recognition(Tjong Kim Sang and De Meulder, 2003), and more recent developments(Ratinov and Roth, 2009; Finkel and Manning, 2009). To the best of our knowledge state-of-the-art features have not been established for labeling all classes of Named Entities in OntoNotes, though Finkel and Manning use the three most common classes and group the others into a *misc* category.

- **character n-grams.** Character sequences of length 1 to 6.
- **case information.** Case of the first, second, and last letter, as well as an indicator for punctuation.
- **lexical information.** The target word, its base form, as well as the same collocations used in $WSD$: $c_{-1,-1}$, $c_{+1,+1}$, $c_{-2,-2}$, $c_{+2,+2}$, $c_{-2,-1}$, $c_{+1,+2}$, $c_{-3,-1}$, $c_{+1,+3}$, $c_{-1,+1}$, $c_{-2,+1}$, $col_{-1,+2}$
- **parts-of-speech.** The part-of-speech for the following words relative to the target word: $p_{-3}, p_{-2}, p_{-1}, p_0, p_{+1}, p_{+2}, p_{+3}$ (0 is the target word).
- **gazetteers.** Mapping of target tokens (or n-grams initiated at the target) to 31 categories based on lists downloaded from Ratinov & Roth (2009).
- **cluster membership.** Mapping of words to Brown (1992) clusters (also downloaded from Ratinov & Roth) based on these positions relative to the target word: $bc_{-2}, bc_{-1}, bc_0, bc_{+1}, bc_{+2}$.
- **selectors.** Selectors were included for the target word as in $WSD$.

Table 8 shows the results for named entity classification. Here, we used one classifier and many potential labels, and thus the most frequent class accuracy is very low, corresponding to the prediction of *organization* for each instance. The inclusion of selectors as features increased the accuracy of our named entity classification system, with a significant 9.3% error reduction. These results, combined with the extensive $WSD$ tests lead us to believe that selectors can also be used successfully as features for many tasks requiring contextual information, such as *prepositional phrase attachment* or *semantic role labeling*, could also benefit from the inclusion of selectors as features.

[5]The Pearson correlation between frequencies of each entity type in our sample versus all instances are 0.982 and 0.945 for the *training* and *test* sets respectively.

| *bill-n.1* | *bill-n.2* | *bill-n.3* | *occur-v.1* | *occur-v.2* | *occur-v.3* |
|---|---|---|---|---|---|
| bill | bill | **market** | be | go | go |
| it | **staff** | system | **happen** | get | **look** |
| **legislation** | system | **paper** | occur | come | **break** |
| system | **money** | **note** | go | have | **remove** |
| **program** | **time** | bill | **take** | **try** | **find** |
| **law** | it | **bond** | work | **lead** | get |
| **plan** | **tax** | **stock** | come | **listen** | **place** |
| **you** | **work** | **debt** | **see** | work | **keep** |
| **measure** | **rent** | **rate** | have | be | **stick** |
| **project** | **tuition** | **report** | **change** | **belong** | **stop** |

Table 9: The ten most common selectors for each sense of the noun 'bill' and the verb 'occur'. Top selectors which are unique to each sense are emboldened.

## 5 Discussion: On the Robustness of Selectors

In the previous section we saw that adding selectors to a standard feature space increases classifier accuracy. In this section, we discuss this improvement by examining the values of features extracted for instances in the SemEval-2007 experimental corpus. Particularly, we endeavor to show that selectors contribute robustness to the *WSD* feature space through an abstraction of context that distinguishes senses of words.

The idea of abstracting context is based on the notion that contexts which realize words of similar meaning have similar selectors. Consider the selectors for senses of both words in Table 9: 'bill' and 'occur'. We see that each of the sets of selectors varies depending on the sense of each word. Furthermore, though coarse-grained, one may even infer the sense of each word by considering its most common selectors; they should be similar to the sense.

For the the supervised classifier, selectors are an encoding or abstraction of context to help identify each sense with no need for concept similarity judgments. For example, both sentences below were annotated incorrectly without selectors, but correctly with selectors.

1. *Polls show wide, generalized support for some vague concept of service, but the **bill** now under discussion lacks any passionate public backing.*
2. *Emerson, in his lecture, refers to the "startling experience which almost every person confesses in daylight, that particular passages of conversation and action have **occurred** to him in the same order before, whether dreaming or waking, a suspicion that they have been with precisely these persons in precisely this room, and heard precisely this dialogue, at some former hour, they know not when".*

For sentence 1, selectors of *bill-n.1* seem to best match the instance's local context (i.e. one can imagine inserting selectors from *bill-n.1* in place of 'bill' more easily than selectors from other senses of bill). Though the training set never contained the exact context "...but the ___ now under..", it did produce selectors which match this context. In sentence 2 the immediate context before and after the target word seem contradictory: "... action have **occurred**..." implies *occur-v.1* (to "happen or take place"), while "... **occurred** to him ..." implies *occur-v.1* (to "come to mind"). However, when considering the whole context *occur-v.2* fits best, and in fact, the selectors for this instance match with many of the most frequent selectors for *occur-v.2* such as 'belong', 'lead', 'listen', and 'try'.

## 5.1 Extensions

We presented evidence that selectors may benefit other lexico-semantic classification tasks in section 4.3.5. Here, we discuss a few extensions to our selector acquisition approach that we believe could bring about further improvements in accuracy. The primary reason we chose to use n-grams as a source for selectors is because the Web search engines that support wildcard search (Yahoo and Google), which is necessary for efficient selector acquisition, no longer support APIs which return all matches. However, because our n-grams were restricted to the order of 5 tokens, the size of local context is limited.

To allow one to search with larger local context, a couple more advanced approaches might be employed. One solution is to use non-wildcard Web search queries (The still-supported Microsoft API could handle this) where candidate selectors are inserted in place of the wildcard. Because this would result in an expensive number of Web queries, one could limit candidates to selectors found through the web-scale 5-grams corpora. Another idea might be to use a smaller corpus than the Web where it is practical to base selectors on grammatical or dependency relationships. A similar approach was done by Lin (1997) without supervision. One can now produce dependency parses over much larger corpora. This would allow one to focus on the important constituents in context as well as capture long distance relationships. Still, part of the attractiveness of web-scale n-grams for selectors is the simplicity. Should our n-gram selectors not contain sufficient local context, one would expect selectors to be ineffective as a type of feature. We found that is not the case.

# 6 Conclusion

We introduced a novel method for increasing the informative value of a supervised disambiguation set of features by leveraging large unannotated corpora to encode an abstraction of local context via *selectors*. When tested over SemEval-2007 Task 17 and Senseval-3 English Lexical Sample, we found that word sense disambiguation classifiers utilizing selectors performed significantly better than those without. The improvements from selectors come free of any annotation cost, requiring only a web-scale n-gram collection. We believe other tasks, such as *prepositional phrase attachment* or *semantic role labeling*, could also benefit from the inclusion of selectors as features.


# Acknowledgments

Support for this research was provided by the Robert Wood Johnson Foundation's Pioneer Portfolio, through a grant to Martin Seligman, "Exploring Concepts of Positive Health."
This work supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number IARPA W911NF-12-C-0023. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.
Fernando Gomez's research was supported in part by the NASA Engineering and Safety Center under Grant/Cooperative Agreement NNX08AJ98A.

## References


Agirre, E., López de Lacalle, O., Fellbaum, C., Hsieh, S., Tesconi, M., Monachini, M., Vossen, P., and Segers, R. (2010). Semeval-2010 task 17: All-words word sense disambiguation on a specific domain. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 75–80, Uppsala, Sweden.

Bergsma, S., Lin, D., and Goebel, R. (2009). Web-scale n-gram models for lexical disambiguation. In *International Joint Conference on Artificial Intelligence*, pages 1507–1512.

Bergsma, S., Pitler, E., and Lin, D. (2010). Creating robust supervised classifiers via web-scale n-gram data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 865–874, Uppsala, Sweden.

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479.

Cai, J. F., Lee, W. S., and Teh, Y. W. (2007). NUS-ML: Improving word sense disambiguation using topic features. In *Proceedings of the International Workshop on Semantic Evaluations*, volume 4.

Cárcamo, J., Gelbukh, A., and Calvo, H. (2008). An innovative two-stage wsd unsupervised method. *Procesamiento del lenguaje natural*, 40:99–105.

Chan, Y. S., Ng, H. T., and Zhong, Z. (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.

Clear, J. H. (1993). The british national corpus. *The digital word: text-based computing in the humanities*, pages 163–187.

Dligach, D. and Palmer, M. (2008). Novel semantic features for verb sense disambiguation. In *Proceedings of ACL-08: HLT, Short Papers*, pages 29–32, Columbus, Ohio. Association for Computational Linguistics.

Downey, D., Broadhead, M., and Etzioni, O. (2007). Locating complex named entities in web text. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 2733–2739.

Finkel, J. R. and Manning, C. D. (2009). Joint parsing and named entity recognition. In *Proceedings of the North American Association of Computational Linguistics (NAACL 2009)*.

Huang, J., Gao, J., Miao, J., Li, X., Wang, K., and Behr, F. (2010). Exploring web scale language models for search query processing. In *19th International World Wide Web Conference (WWW-2010)*, Raleigh, NC.

Lee, Y. K. and Ng, H. T. (2002). An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 41–48, Morristown, NJ, USA.

Lin, D. (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.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL 98*, pages 768–774, Montreal, Canada. Morgan Kaufmann.

Lin, D., Church, K., Ji, H., Sekine, S., Yarowsky, D., Bergsma, S., Patil, K., Pitler, E., et al. (2010). New tools for web-scale n-grams. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta.

McCarthy, D., Koeling, R., Weeds, J., and Carroll, J. (2004). Finding predominant word senses in untagged text. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, pages 280–287, Barcelona, Spain. Association for Computational Linguistics.

Mihalcea, R. (2004). Co-training and self-training for word sense disambiguation. In *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning*, pages 33–40.

Mihalcea, R. (2007). Using wikipedia for automatic word sense disambiguation. In *North American Chapter of the Association for Computational Linguistics (NAACL 2007)*.

Mihalcea, R., Chklovski, T., and Kilgarriff, A. (2004). The senseval-3 english lexical sample task. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 25–28, Barcelona, Spain. Association for Computational Linguistics.

Mihalcea, R. and Moldovan, D. I. (1999). An automatic method for generating sense tagged corpora. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI-99)*, pages 461–466.

Miller, G., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. (1993). Five papers on wordnet. Technical report, Princeton University.

Navigli, R., Litkowski, K. C., and Hargraves, O. (2007). Semeval-2007 task 07: Coarse-grained english all-words task. In *Proceedings of SemEval-2007*, pages 30–35, Prague, Czech Republic.

Pantel, P. and Lin, D. (2002). Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '02)*, pages 613–619, New York, NY, USA. ACM Press.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and E., D. (2011). Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830.

Pereira, F., Tishby, N., and Lee, L. (1993). Distributional clustering of english words. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 183–190, Stroudsburg, PA, USA.

Pham, T. P., Ng, H. T., and Lee, W. S. (2005). Word sense disambiguation with semi-supervised learning. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 3*, AAAI'05, pages 1093–1098.

Pradhan, S. S., Loper, E., Dligach, D., and Palmer, M. (2007). Semeval-2007 task 17: English lexical sample, srl and all words. In *Proceedings of SemEval-2007*, pages 87–92.

Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155.

Resnik, P. (1997). Selectional preference and sense disambiguation. In *Proceedings of the ANLP Workshop: Tagging Text with Lexical Semantics*, Washington, DC, USA.

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Schwartz, H. A. and Gomez, F. (2008). Acquiring knowledge from the web to be used as selectors for noun sense disambiguation. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, Manchester, England.

Schwartz, H. A. and Gomez, F. (2009). Using web selectors for the disambiguation of all words. In *Proceedings of the NAACL-2009 Workshop on Semantic Evaluations (SEW-2009)*, Boulder, Colorado.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Daelemans, W. and Osborne, M., editors, *Proceedings of CoNLL-2003*, pages 142–147, Edmonton, Canada.

Weischedel, R., Palmer, M., Marcus, M., Hovy, E., Pradhan, S., Ramshaw, L., Xue, N., Taylor, A., Kaufman, J., Franchini, M., et al. (2011). Ontonotes release 4.0. In *LDC2011T03, Philadelphia, Penn.* Linguistic Data Consortium.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196.

Yuret, D. (2007). KU: Word sense disambiguation by substitution. In *Proceedings of SemEval-2007*, pages 207–214, Prague, Czech Republic.

Zhong, Z., Ng, H. T., and Chan, Y. S. (2008). Word sense disambiguation using OntoNotes: An empirical study. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1002–1010, Honolulu, Hawaii.

Zhu, X. (2008). Semi-supervised learning literature survey. technical report.