Inducing Ontological Co-occurrence Vectors

Patrick Pantel
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
pantel@isi.edu

Abstract

In this paper, we present an unsupervised methodology for propagating lexical co-occurrence vectors into an ontology such as WordNet. We evaluate the framework on the task of automatically attaching new concepts into the ontology. Experimental results show 73.9% attachment accuracy in the first position and 81.3% accuracy in the top-5 positions. This framework could potentially serve as a foundation for ontologizing lexical-semantic resources and assist the development of other large-scale and internally consistent collections of semantic information.

1 Introduction

Despite considerable effort, there is still today no commonly accepted semantic corpus, semantic framework, notation, or even agreement on precisely which aspects of semantics are most useful (if at all). We believe that one important reason for this rather startling fact is the absence of truly wide-coverage semantic resources.

Recognizing this, some recent work on wide coverage term banks, like WordNet (Miller 1990) and CYC (Lenat 1995), and annotated corpora, like FrameNet (Baker et al. 1998), Propbank (Kingsbury et al. 2002) and Nombank (Meyers et al. 2004), seeks to address the problem. But manual efforts such as these suffer from two drawbacks: they are difficult to tailor to new domains, and they have internal inconsistencies that can make automating the acquisition process difficult.

In this work, we introduce a general framework for inducing co-occurrence feature vectors for nodes in a WordNet-like ontology. We believe that this framework will be useful for a variety of applications, including adding additional semantic information to existing semantic term banks by disambiguating lexical-semantic resources.

Ontologizing semantic resources

Recently, researchers have applied text- and web-mining algorithms for automatically creating lexical semantic resources like similarity lists (Lin 1998), semantic lexicons (Riloff and Shepherd 1997), hyponymy lists (Shinzato and Torisawa 2004; Pantel and Ravichandran 2004), part-whole lists (Girgu et al. 2003), and verb relation graphs (Chklovski and Pantel 2004). However, none of these resources have been directly linked into an ontological framework. For example, in VERBOCEAN (Chklovski and Pantel 2004), we find the verb relation "to surpass *is-stronger-than* to hit", but it is not specified that it is the achieving sense of *hit* where this relation applies.

We term *ontologizing* a lexical-semantic resource as the task of sense disambiguating the resource. This problem is different but not orthogonal to word-sense disambiguation. If we could disambiguate large collections of text with high accuracy, then current methods for building lexical-semantic resources could easily be applied to ontologize them by treating each word's senses as separate words. Our method does not require the disambiguation of text. Instead, it relies on the principle of distributional similarity and that polysemous words that are similar in one sense are dissimilar in their other senses.

Given the enriched ontologies produced by our method, we believe that ontologizing lexical-semantic resources will be feasible. For example, consider the example verb relation "to surpass *is-stronger-than* to hit" from above. To disambiguate the verb *hit*, we can look at all other verbs that *to surpass* is stronger than (for example, in VERBOCEAN, "to surpass *is-stronger-than* to overtake" and "to surpass *is-stronger-than* to equal"). Now, we can simply compare the lexical co-occurrence vectors of *overtake* and *equal* with the ontological feature vectors of the senses of *hit* (which are induced by our framework). The sense whose feature vector is most similar is selected.

It remains to be seen in future work how well this approach performs on ontologizing various semantic resources. In this paper, we focus on the general framework for inducing the ontological co-occurrence vectors and we apply it to the task of linking new terms into the ontology.

2 Relevant work

Our framework aims at enriching WordNet-like ontologies with syntactic features derived from a non-annotated corpus. Others have also made significant additions to WordNet. For example, in eXtended WordNet (Harabagiu et al. 1999), the rich glosses in WordNet are enriched by disambiguating the nouns, verbs, adverbs, and adjectives with synsets. Another work has enriched WordNet synsets with topically related words extracted from the Web (Agirre et al. 2001). While this method takes advantage of the redundancy of the web, our source of information is a local document collection, which opens the possibility for domain specific applications.

Distributional approaches to building semantic repositories have shown remarkable power. The underlying assumption, called the Distributional Hypothesis (Harris 1985), links the semantics of words to their lexical and syntactic behavior. The hypothesis states that words that occur in the same contexts tend to have similar meaning. Researchers have mostly looked at representing words by their surrounding words (Lund and Burgess 1996) and by their syntactical contexts (Hindle 1990; Lin 1998). However, these representations do not distinguish between the different senses of words. Our framework utilizes these principles and representations to induce disambiguated feature vectors. We describe these representations further in Section 3.

In supervised word sense disambiguation, senses are commonly represented by their surrounding words in a sense-tagged corpus (Gale et al. 1991). If we had a large collection of sense-tagged text, then we could extract disambiguated feature vectors by collecting co-occurrence features for each word sense. However, since there is little sense-tagged text available, the feature vectors for a random WordNet concept would be very sparse. In our framework, feature vectors are induced from much larger untagged corpora (currently 3GB of newspaper text).

Another approach to building semantic repositories is to collect and merge existing ontologies. Attempts to automate the merging process have not been particularly successful (Knight and Luk 1994; Hovy 1998; Noy and Musen 1999). The principal problems of partial and unbalanced coverage and of inconsistencies between ontologies continue to hamper these approaches.

3 Resources

The framework we present in Section 4 propagates any type of lexical feature up an ontology. In previous work, lexicals have often been represented by proximity and syntactic features. Consider the following sentence:

```
The tsunami left a trail of horror.
```

In a proximity approach, a word is represented by a window of words surrounding it. For the above sentence, a window of size 1 would yield two features (*-1:the* and *+1:left*) for the word *tsunami*. In a syntactic approach, more linguistically rich features are extracted by using each grammatical relation in which a word is involved (e.g. the features for *tsunami* are *determiner:the* and *subject-of:leave*).

For the purposes of this work, we consider the propagation of syntactic features. We used Minipar (Lin 1994), a broad coverage parser, to analyze text. We collected the statistics on the grammatical relations (contexts) output by Minipar and used these as the feature vectors. Following Lin (1998), we measure each feature f for a word e not by its frequency but by its pointwise mutual information, mi_{ef}:

Input: A node *n* and a corpus *C*.

Step 1: Termination Condition:

If *n* is a leaf node then assign to *n* its lexical feature vector as described in Section 3.

Step 2: Recursion Step:

For each child *c* of *n*, reecurse on *c* and *C*.
Assign a feature vector to *n* by propagating features from its children.

Output: A feature vector assigned to each node of the tree rooted by *n*.

Figure 1. Divide-and-conquer phase.

$$mi_{ef} = \log \frac{P(e, f)}{P(e) \times P(f)}$$

4 Inducing ontological features

The resource described in the previous section yields lexical feature vectors for each word in a corpus. We term these vectors *lexical* because they are collected by looking only at the lexicals in the text (i.e. no sense information is used). We use the term *ontological feature vector* to refer to a feature vector whose features are for a particular sense of the word.

In this section, we describe our framework for inducing ontological feature vectors for each node of an ontology. Our approach employs two phases. A divide-and-conquer algorithm first propagates syntactic features to each node in the ontology. A final sweep over the ontology, which we call the Coup phase, disambiguates the feature vectors of lexicals (leaf nodes) in the ontology.

4.1 Divide-and-conquer phase

In the first phase of the algorithm, we propagate features up the ontology in a bottom-up approach. Figure 1 gives an overview of this phase.

The termination condition of the recursion is met when the algorithm processes a leaf node. The feature vector that is assigned to this node is an exact copy of the lexical feature vector for that leaf (obtained from a large corpus as described in Section 3). For example, for the two leaf nodes labeled *chair* in Figure 2, we assign to both the same ambiguous lexical feature vector, an excerpt of which is shown in Figure 3.

When the recursion meets a non-leaf node, like *chairwoman* in Figure 2, the algorithm first

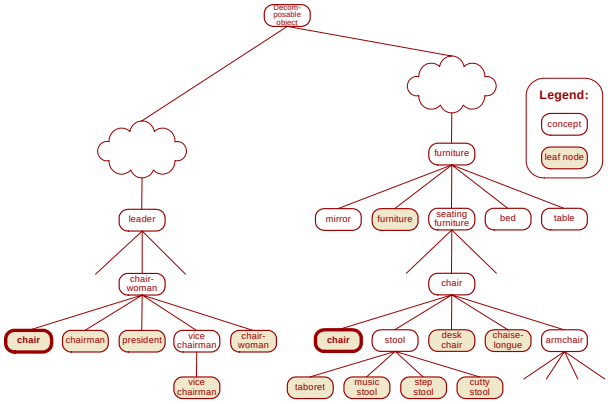

Figure 2. Subtrees of WordNet illustrating two senses of *chair*.

```
"chair"
   conjunction:
       sofa                  77     11.8
       professor             11     6.0
       dining room           2      5.6
       cushion               1      4.5
       council member        1      4.4
       President             9      2.9
       foreign minister      1      2.8
   nominal subject
       Ottoman               8      12.1
       director              22     9.1
       speaker               8      8.6
       Joyner                2      8.22
       recliner              2      7.7
       candidate             1      3.5
```

Figure 3. Excerpt of a lexical feature vector for the word *chair*. Grammatical relations are in italics (*conjunction* and *nominal-subject*). The first column of numbers are frequency counts and the other are mutual information scores. In bold are the features that intersect with the induced ontological feature vector for the parent concept of *chair*'s *chairwoman* sense.

recursively applies itself to each of the node's children. Then, the algorithm selects those features common to its children to propagate up to its own ontological feature vector. The assumption here is that features of other senses of polysemous words will not be propagated since they will not be common across the children. Below, we describe the two methods we used to propagate features: *Shared* and *Committee*.

Shared propagation algorithm

The first technique for propagating features to a concept node *n* from its children *C* is the simplest and scored best in our evaluation (see Section 5.2). The goal is that the feature vector for *n*

represents the general grammatical behavior that its children will have. For example, for the concept node *furniture* in Figure 2, we would like to assign features like *object-of:clean* since mosttypes of furniture can be cleaned. However, even though you can eat on a table, we do not want the feature *on:eat* for the *furniture* concept since we do not eat on *mirrors* or *beds*.

In the *Shared* propagation algorithm, we propagate only those features that are shared by at least t children. In our experiments, we experimentally set $t = min(3, |C|)$.

The frequency of a propagated feature is obtained by taking a weighted sum of the frequency of the feature across its children. Let f_i be the frequency of the feature for child i, let c_i be the total frequency of child i, and let N be the total frequency of all children. Then, the frequency f of the propagated feature is given by:

$$f = \sum_i f_i \times \frac{c_i}{N} \quad (1)$$

Committee propagation algorithm

The second propagation algorithm finds a set of representative children from which to propagate features. Pantel and Lin (2002) describe an algorithm, called Clustering By Committee (CBC), which discovers clusters of words according to their meanings in test. The key to CBC is finding for each class a set of representative elements, called a committee, which most unambiguously describe the members of the class. For example, for the *color* concept, CBC discovers the following committee members:

```
purple, pink, yellow, mauve, turquoise,
beige, fuchsia
```

Words like *orange* and *violet* are avoided because they are polysemous. For a given concept c, we build a committee by clustering its children according to their similarity and then keep the largest and most interconnected cluster (see Pantel and Lin (2002) for details).

The propagated features are then those that are shared by at least two committee members. The frequency of a propagated feature is obtained using Eq. 1 where the children i are chosen only among the committee members.

Generating committees using CBC works best for classes with many members. In its original application (Pantel and Lin 2002), CBC discovered a flat list of coarse concepts. In the finer grained concept hierarchy of WordNet, there are many fewer children for each concept so we expect to have more difficulty finding committees.

Input:	A node n and the enriched ontology O output from the algorithm in Figure 1.
Step 1:	If n is not a leaf node then return.
Step 2:	Remove from n's feature vector all features that intersect with the feature vector of any of n's other senses' parent concepts, but are not in n's parent concept feature vector.
Output:	A disambiguated feature vector for each leaf node n.

Figure 4. Coup phase.

4.2 Coup phase

At the end of the *Divide-and-conquer* phase, the non-leaf nodes of the ontology contain disambiguated features[1]. By design of the propagation algorithm, each concept node feature is shared by at least two of its children. We assume that two polysemous words, w_1 and w_2, that are similar in one sense will be dissimilar in its other senses. Under the distributional hypothesis, similar words occur in the same grammatical contexts and dissimilar words occur in different grammatical contexts. We expect then that most features that are shared between w_1 and w_2 will be the grammatical contexts of their similar sense. Hence, mostly disambiguated features are propagated up the ontology in the *Divide-and-conquer* phase.

However, the feature vectors for the leaf nodes remain ambiguous (e.g. the feature vectors for both leaf nodes labeled *chair* in Figure 2 are identical). In this phase of the algorithm, leaf node feature vectors are disambiguated by looking at the parents of their other senses.

Leaf nodes that are unambiguous in the ontology will have unambiguous feature vectors. For ambiguous leaf nodes (i.e. leaf nodes that have more than one concept parent), we apply the algorithm described in Figure 4. Given a polysemous leaf node n, we remove from its ambiguous

[1] By disambiguated features, we mean that the features are co-occurrences with a particular sense of a word; the features themselves are not sense-tagged.

feature vector those features that intersect with the ontological feature vector of any of its other senses' parent concept but that are not in its own parent's ontological feature vector. For example, consider the *furniture* sense of the leaf node *chair* in Figure 2. After the *Divide-and-conquer* phase, the node *chair* is assigned the ambiguous lexical feature vector shown in Figure 3. Suppose that *chair* only has one other sense in WordNet, which is the *chairwoman* sense illustrated in Figure 2. The features in bold in Figure 3 represent those features of *chair* that intersect with the ontological feature vector of *chairwoman*. In the *Coup* phase of our system, we remove these bold features from the *furniture* sense leaf node *chair*. What remains are features like "*chair and sofa*", "*chair and cushion*", "*Ottoman is a chair*", and "*recliner is a chair*". Similarly, for the *chairwoman* sense of *chair*, we remove those features that intersect with the ontological feature vector of the *chair* concept (the parent of the other *chair* leaf node).

As shown in the beginning of this section, concept node feature vectors are mostly unambiguous after the *Divide-and-conquer* phase. However, the *Divide-and-conquer* phase may be repeated after the *Coup* phase using a different termination condition. Instead of assigning to leaf nodes ambiguous lexical feature vectors, we use the leaf node feature vectors from the *Coup* phase. In our experiments, we did not see any significant performance difference by skipping this extra *Divide-and-conquer* step.

5 Experimental results

In this section, we provide a quantitative and qualitative evaluation of our framework.

5.1 Experimental Setup

We used Minipar (Lin 1994), a broad coverage parser, to parse two 3GB corpora (TREC-9 and TREC-2002). We collected the frequency counts of the grammatical relations (contexts) output by Minipar and used these to construct the lexical feature vectors as described in Section 3.

WordNet 2.0 served as our testing ontology. Using the algorithm presented in Section 4, we induced ontological feature vectors for the noun nodes in WordNet using the lexical co-occurrence features from the TREC-2002 corpus. Due to memory limitations, we were only able to propagate features to one quarter of the ontology. We experimented with both the *Shared* and *Committee* propagation models described in Section 4.1.

5.2 Quantitative evaluation

To evaluate the resulting ontological feature vectors, we considered the task of attaching new nodes into the ontology. To automatically evaluate this, we randomly extracted a set of 1000 noun leaf nodes from the ontology and accumulated lexical feature vectors for them using the TREC-9 corpus (a separate corpus than the one used to propagate features, but of the same genre). We experimented with two test sets:

- *Full*: The 424 of the 1000 random nodes that existed in the TREC-9 corpus
- *Subset*: Subset of *Full* where only nodes that do not have concept siblings are kept (380 nodes).

For each random node, we computed the similarity of the node with each concept node in the ontology by computing the cosine of the angle (Salton and McGill 1983) between the lexical feature vector of the random node e_i and the ontological feature vector of the concept nodes e_j:

$$sim(e_i, e_j) = \frac{\sum_f mi_{e_i f} \times mi_{e_j f}}{\sqrt{\sum_f mi_{e_i f}^2 \times \sum_f mi_{e_j f}^2}}$$

We only kept those similar nodes that had a similarity above a threshold σ. We experimentally set σ = 0.1.

Top-*K* accuracy

We collected the top-*K* most similar concept nodes (attachment points) for each node in the test sets and computed the accuracy of finding a correct attachment point in the top-*K* list. Table 1 shows the result.

We expected the algorithm to perform better on the *Subset* data set since only concepts that have exclusively lexical children must be considered for attachment. In the *Full* data set, the algorithm must consider each concept in the ontology as a potential attachment point. However, considering the top-5 best attachments, the algorithm performed equally well on both data sets.

The *Shared* propagation algorithm performed consistently slightly better than the *Committee* method. As described in Section 4.1, building a

Table 1. Correct attachment point in the top-*K* attachments (with 95% conf.)

K	*Shared (Full)*	*Committee (Full)*	*Shared (Subset)*	*Committee (Subset)*
1	73.9% ± 4.5%	72.0% ± 4.9%	77.4% ± 3.6%	76.1% ± 5.1%
2	78.7% ± 4.1%	76.6% ± 4.2%	80.7% ± 4.0%	79.1% ± 4.5%
3	79.9% ± 4.0%	78.2% ± 4.2%	81.2% ± 3.9%	80.5% ± 4.8%
4	80.6% ± 4.1%	79.0% ± 4.0%	81.5% ± 4.1%	80.8% ± 5.0%
5	81.3% ± 3.8%	79.5% ± 3.9%	81.7% ± 4.1%	81.3% ± 4.9%

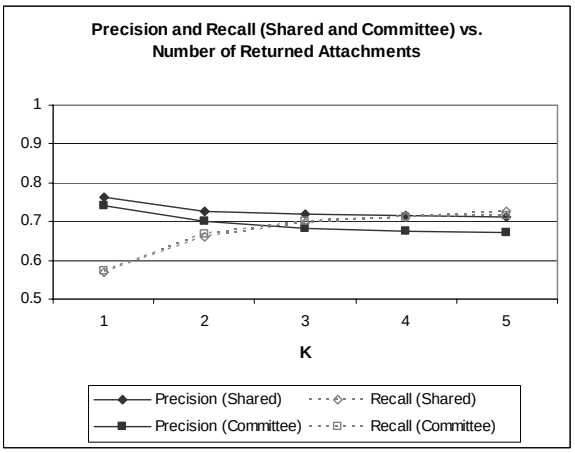

Figure 5. Attachment precision and recall for the *Shared* and *Committee* propagation methods when returning at most *K* attachments (on the *Full* set).

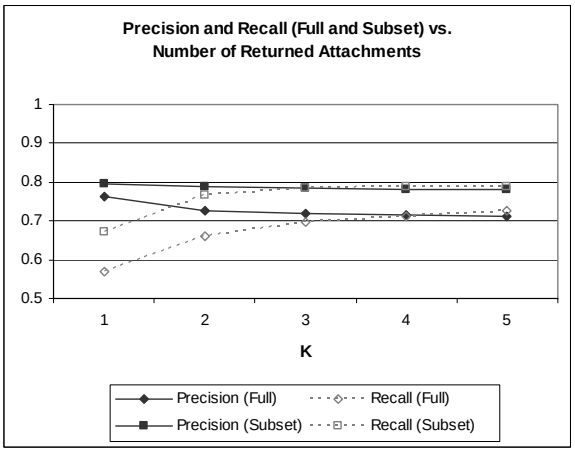

Figure 6. Attachment precision and recall for the *Full* and *Subset* data sets when returning at most *K* attachments (using the *Shared* propagation method).

committee performs best for concepts with many children. Since many nodes in WordNet have few direct children, the *Shared* propagation method is more appropriate. One possible extension of the *Committee* propagation algorithm is to find committee members from the full list of descendants of a node rather than only its immediate children.

Precision and Recall

We computed the precision and recall of our system on varying numbers of returned attachments. Figure 5 and Figure 6 show the attachment precision and recall of our system when the maximum number of returned attachments ranges between 1 and 5. In Figure 5, we see that the *Shared* propagation method has better precision than the *Committee* method. Both methods perform similarly on recall. The recall of the system increases most dramatically when returning two attachments without too much of a hit on precision. The low recall when returning only one attachment is due to both system errors and also to the fact that many nodes in the hierarchy are polysemous. In the next section, we discuss further experiments on polysemous nodes. Figure 6 illustrates the large difference on both precision and recall when using the simpler *Subset* data set. All 95% confidence bounds in Figure 5 and Figure 6 range between ±2.8% and ±5.3%.

Polysemous nodes

84 of the nodes in the Full data set are polysemous (they are attached to more than one concept node in the ontology). On average, these nodes have 2.6 senses for a total of 219 senses. Figure 7 compares the precision and recall of the system on all nodes in the *Full* data set vs. the 84 polysemous nodes. The 95% confidence intervals range between ±3.8% and ±5.0% for the *Full* data set and between ±1.2% and ±9.4% for the polysemous nodes. The precision on the polysemous nodes is consistently better since these have more possible correct attachments.

Clearly, when the system returns at most one or two attachments, the recall on the polysemous nodes is lower than on the *Full* set. However, it is interesting to note that recall on the polysemous nodes equals the recall on the *Full* set after *K*=3.

5.3 Qualitative evaluation

Inspection of errors revealed that the system often makes plausible attachments. Table 2 shows some example errors generated by our system. For the word *arsenic*, the system attached it to the concept *trioxide*, which is the parent of the correct attachment.

The system results may be useful to help validate the ontology. For example, for the word *law*, the system attached it to the *regulation* (as an organic process) and *ordinance* (legislative act) concepts. According to WordNet, *law* has seven possible attachment points, none of which are a legislative act. Hence, the system has found that in the TREC-9 corpus, the word *law* has a sense of *legislative act*. Similarly, the system discovered the *symptom* sense of *vomiting*.

The system discovered a potential anomaly in WordNet with the word *slob*. The system classified *slob* as follows:

`fool → simpleton → someone`

whereas WordNet classifies it as:

`vulgarian → unpleasant person → unwelcome person → someone`

The ontology could use this output to verify if *fool* should link in the *unpleasant person* subtree.

Capitalization is not very trustworthy in large collections of text. One of our design decisions was to ignore the case of words in our corpus, which in turn caused some errors since WordNet is case sensitive. For example, the lexical node *Munch* (Norwegian artist) was attached to the *munch* concept (food) by error because our system accumulated all features of the word *Munch* in text regardless of its capitalization.

6 Discussion

One question that remains unanswered is how clean an ontology must be in order for our methodology to work. Since the structure of the ontology guides the propagation of features, a very noisy ontology will result in noisy feature vectors. However, the framework is tolerant to some amount of noise and can in fact be used to correct some errors (as shown in Section 5.3).

We showed in Section 1 how our framework can be used to disambiguate lexical-semantic resources like hyponym lists, verb relations, and

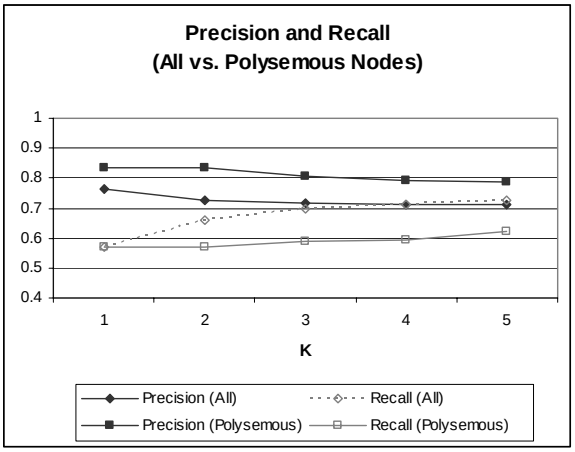

Figure 7. Attachment precision and recall on the *Full* set vs. the polysemous nodes in the *Full* set when the system returns at most *K* attachments.

unknown words or terms. Other avenues of future work include:

Adapting/extending existing ontologies
It takes a large amount of time to build resources like WordNet. However, adapting existing resources to a new corpus might be possible using our framework. Once we have enriched the ontology with features from a corpus, we can rearrange the ontological structure according to the inter-conceptual similarity of nodes. For example, consider the word *computer* in WordNet, which has two senses: *a*) a machine; and *b*) a person who calculates. In a computer science corpus, sense *b*) occurs very infrequently and possibly a new sense of computer (e.g. *a processing chip*) occurs. A system could potentially remove sense *b*) since the similarity of the other children of *b*) and *computer* is very low. It could also uncover the new *processing chip* sense by finding a high similarity between *computer* and the *processing chip* concept.

Validating ontologies
This is a holy grail problem in the knowledge representation community. As a small step, our framework can be used to flag potential anomalies to the knowledge engineer.

What makes a *chair* different from a *recliner*?
Given an enriched ontology, we can remove from the feature vectors of *chair* and *recliner* those features that occur in their parent *furniture* concept. The features that remain describe their different syntactic behaviors in text.

Table 2. Example attachment errors by our system.

Node	*System Attachment*	*Correct Attachment*
arsenic*	trioxide	arsenic OR element
law	regulation	law OR police OR …
Munch†	munch	Munch
slob	fool	slob
vomiting	fever	emesis

* the system's attachment was a parent of the correct attachment.
† error due to case mix-up (our algorithm does not differentiate between case).

7 Conclusions

We presented a framework for inducing ontological feature vectors from lexical co-occurrence vectors. Our method does not require the disambiguation of text. Instead, it relies on the principle of distributional similarity and the fact that polysemous words that are similar in one sense tend to be dissimilar in their other senses. On the task of attaching new words to WordNet using our framework, our experiments showed that the first attachment has 73.9% accuracy and that a correct attachment is in the top-5 attachments with 81.3% accuracy.

We believe this work to be useful for a variety of applications. Not only can sense selection tasks such as word sense disambiguation, parsing, and semantic analysis benefit from our framework, but more inference-oriented tasks such as question answering and text summarization as well. We hope that this work will assist with the development of other large-scale and internally consistent collections of semantic information.

References

Agirre, E.; Ansa, O.; Martinez, D.; and Hovy, E. 2001. Enriching WordNet concepts with topic signatures. In *Proceedings of the NAACL workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations.* Pittsburgh, PA.

Baker, C.; Fillmore, C.; and Lowe, J. 1998. The Berkeley FrameNet project. In *Proceedings of COLING-ACL*. Montreal, Canada.

Chklovski, T., and Pantel, P. VERBOCEAN: Mining the Web for Fine-Grained Semantic Verb Relations. In *Proceedings of EMNLP-2004*. pp. 33-40. Barcelona, Spain.

Gale, W.; Church, K.; and Yarowsky, D. 1992. A method for disambiguating word senses in a large corpus. *Computers and Humanities*, 26:415-439.

Girju, R.; Badulescu, A.; and Moldovan, D. 2003. Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of HLT/NAACL-03*. pp. 80-87. Edmonton, Canada.

Harabagiu, S.; Miller, G.; and Moldovan, D. 1999. WordNet 2 - A Morphologically and Semantically Enhanced Resource. In *Proceedings of SIGLEX-99*. pp.1-8. University of Maryland.

Harris, Z. 1985. Distributional structure. In: Katz, J. J. (ed.) *The Philosophy of Linguistics*. New York: Oxford University Press. pp. 26-47.

Hovy, E. 1998. Combining and standardizing large-scale, practical ontologies for machine translation and other uses. In *Proceedings LREC-98*. pp. 535-542. Granada, Spain.

Hindle, D. 1990. Noun classification from predicate-argument structures. In *Proceedings of ACL-90*. pp. 268-275. Pittsburgh, PA.

Kingsbury, P; Palmer, M.; and Marcus, M. 2002. Adding semantic annotation to the Penn TreeBank. In *Proceedings of HLT-2002*. San Diego, California.

Knight, K. and Luk, S. K. 1994. Building a large-scale knowledge base for machine translation. In *Proceedings of AAAI-1994*. Seattle, WA.

Lenat, D. 1995. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33-38.

Lin, D. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING/ACL-98*. pp. 768-774. Montreal, Canada.

Lin, D. 1994. Principar - an efficient, broad-coverage, principle-based parser. *Proceedings of COLING-94*. pp. 42-48. Kyoto, Japan.

Lund, K. and Burgess, C. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, and Computers*, 28:203-208.

Meyers, A.; Reeves, R.; Macleod, C.; Szekely, R.; Zielinska, V.; Young, B.; and Grishman, R. Annotating noun argument structure for NomBank. In *Proceedings of LREC-2004*. Lisbon, Portugal.

Miller, G. 1990. WordNet: An online lexical database. *International Journal of Lexicography*, 3(4).

Noy, N. F. and Musen, M. A. 1999. An algorithm for merging and aligning ontologies: Automation and tool support. In *Proceedings of the Workshop on Ontology Management (AAAI-99)*. Orlando, FL.

Pantel, P. and Lin, D. 2002. Discovering Word Senses from Text. In *Proceedings of SIGKDD-02*. pp. 613-619. Edmonton, Canada.

Riloff, E. and Shepherd, J. 1997. A corpus-based approach for building semantic lexicons. In *Proceedings of EMNLP-1997*.

Salton, G. and McGill, M. J. 1983. *Introduction to Modern Information Retrieval*. McGraw Hill.

Shinzato, K. and Torisawa, K. 2004. Acquiring hyponymy relations from web documents. In *Proceedings of HLT-NAACL-2004*. pp. 73-80. Boston, MA.