

# Using Selectional Profile Distance to Detect Verb Alternations

Vivian Tsang and Suzanne Stevenson

Department of Computer Science

University of Toronto

{vyctsang, suzanne}@cs.toronto.edu

## Abstract

We propose a new method for detecting verb alternations, by comparing the probability distributions over WordNet classes occurring in two potentially alternating argument positions. Existing distance measures compute only the distributional distance, and do not take into account the semantic similarity between WordNet senses across the distributions. Our method compares two probability distributions over WordNet by measuring the semantic distance of the component nodes, weighted by their probability. To incorporate semantic similarity, we calculate the (dis)similarity between two probability distributions as a weighted distance “travelled” from one to the other through the WordNet hierarchy. We evaluate the measure on the causative alternation, and find that overall it outperforms existing distance measures.

## 1 Detecting Verb Alternations

Although patterns of verb alternations, as in (1) and (2), may appear to be “mere” syntactic variation, the ability of a verb to alternate has been shown to be highly related to its semantic properties.

1. The sun melted the snow./The snow melted.
2. Kiva ate his lunch./Kiva ate./\*His lunch ate.

For example, *melt* in (1) undergoes a causative alternation in which the transitive form is related to the intransitive by the introduction of a Causal Agent (*the sun*) into the event structure. The verb *eat* in (2), like *melt*, allows both transitive and intransitive forms, but these are related by the unspecified object alternation, as opposed to causativization.

Based largely on the influence of Levin (1993), it has become widely accepted that alternations such as these can serve as a basis for the formation of semantic classes of verbs. Correspondingly, the relation between alternation patterns and meaning is a key focus in the computational study of the lexical semantics of verbs (e.g., Allen, 1997; Dang et al., 2000; Dorr and Jones, 2000; Merlo and Stevenson, 2001; Schulte im Walde and Brew, 2002; Tsang et al., 2002). Furthermore, we note that recent work indicates that verb alternations may also play a role in automatic processing of language for applied tasks, such as question-answering (Katz et al., 2001), detection of text relations (Teufel, 1999), and determination of verb-particle constructions (Bannard, 2002).

The theoretical and practical implications of alternations mean that it is important to identify verbs which undergo an alternation, and to discover the range of alternations. Manual annotation of verbs is labour intensive, and new verbs (or new uses of known verbs) may be encountered in any given domain. In response, some researchers have begun to investigate ways to detect alternations automatically in a corpus. Some of this work has focused on subcategorization patterns as the clear syntactic cue to an alternation (Lapata, 1999; Lapata and Brew, 1999; Schulte im Walde and Brew, 2002). Other work has observed, however, that detecting an alternation involves more than observing the use of particular subcategorizations—it must also be determined whether the semantic arguments are mapped to the appropriate positions.<sup>1</sup>

To address this issue, it has been suggested that, if a verb participates in an alternation, then there should be similarity in the kinds of nouns that show up in the syn-

<sup>1</sup>For example, *melt* (as in (1) above) undergoes a causative alternation because the Theme argument that surfaces as subject of the intransitive surfaces as object of the transitive, with the addition of a Causal Agent as the subject of the latter. It is not the case that any optionally intransitive verb undergoes this alternation, as shown by *eat* in (2).

tactic positions (or slots) that alternate—such as *snow* occurring as intransitive subject and transitive object in the causative alternation in (1) (Merlo and Stevenson, 2001; McCarthy, 2000). As a cue to this alternation, Merlo and Stevenson (2001) create a bag of head nouns for each of the two potentially alternating slots, and compare them. In contrast to comparing head nouns directly, McCarthy (2000) instead compares the selectional preferences for each of the two slots (captured by a probability distribution over WordNet). This approach thereby generalizes over the compared nouns, increasing performance over a method similar to that of Merlo and Stevenson.

In our work, we have developed a new method for comparing WordNet probability distributions, called “selectional profile distance” (SPD), which combines the benefits of each of the above approaches for detecting alternations. The method used by Merlo and Stevenson (2001) has the advantage of directly capturing similarity between slots (in terms of use of identical nouns [lemmas]), but fails to generalize over the nouns, lending itself to sparse data problems. The approach of McCarthy (2000), on the other hand, addresses the generalization problem by comparing probability distributions over WordNet. However, her comparison measure abstracts over distances between nodes (classes of nouns) in WordNet: it rewards probability mass that occurs in the same subtree across two distributions, but does not take into account the distance between the classes that carry the probability mass. Thus, this approach only captures similarity among the noun arguments across slots at a very coarse level. Our new SPD method integrates a comparison of probability distributions over WordNet with a node similarity measure, successfully capturing both of the advantageous properties of generalization and word (class) similarity. SPD thus enables us to calculate a meaningful similarity measure over the patterns of classes of nouns across two syntactic slots.

Our evaluation of the SPD measure for alternation detection also covers some interesting experimental conditions that have not been explored previously. For comparison to previous methods, we investigate these issues in the context of classifying verbs according to whether they undergo the causative alternation. We experiment with randomly selected verbs, for both our alternating and non-alternating (filler) classes, and use both relatively homogeneous and heterogeneous sets of filler verbs. We find that our method performs about the same on each set, indicating that it is insensitive to variation in the filler verbs. Moreover, we experiment with equal numbers of verbs in different frequency bands, and show that splitting verbs into high and low frequency (of slot occurrence) can improve performance. By classifying the high and low frequency verbs separately, our method achieves an accuracy of 70% overall on unseen test verbs, in a

task with a baseline of 50%. (For comparison, McCarthy (2000) achieves 73% on her set of hand-selected verbs, but our implementation of her method yields much lower performance on our randomly selected test verbs.)

In the next section, we present background work on capturing selectional preferences in WordNet, and on using them to detect alternations. In Section 3, we describe our new SPD measure, and show how it captures both the general differences between WordNet probability distributions, as well as the fine-grained semantic distances between the nodes that comprise them. Section 4 presents our corpus methodology and experimental set-up. In Section 5, we compare SPD to a range of distance measures, and evaluate the different effects of our experimental factors, such as the precise distance functions we use in SPD and the division of our verbs into frequency bands. We summarize our findings in Section 6 and point to directions in our on-going work.

## 2 The Use of Selectional Preferences

Selectional preference refers to the general notion of how much a verb favours (or disfavors) a particular noun as a semantic argument. For example, informally we would say that *eat* has a strong selectional preference for nouns of type food as its Theme argument. Formalization of this notion has been difficult, but several computational methods have now been proposed that capture selectional preference of a verb as a probability distribution over the WordNet hierarchy (Resnik, 1993; Li and Abe, 1998; Clark and Weir, 2002).<sup>2</sup> The key task that each of these proposals address is how to generalize appropriately from counts of observed nouns in the relevant verb argument position (in a corpus), to a probabilistic representation of selectional strength over classes. We will refer in the remainder of the paper to such a probability distribution over WordNet as a “selectional profile.”

As mentioned above, McCarthy (2000) suggested the use of selectional profiles to capture generalizations over argument slots, so that two argument slots could be effectively compared for detecting alternations. After extracting the argument heads of the target slots of each verb (e.g., the intransitive subject and the transitive object for the causative alternation), she then determined their selectional profiles using a minimum description length tree cut model (Li and Abe, 1998).<sup>3</sup> The two slot profiles were compared using skew divergence (a variant of

<sup>2</sup>Resnik’s proposed measure is not actually a probability distribution, but a difference between probability distributions.

<sup>3</sup>A tree cut for tree  $T$  is a set of nodes  $C$  in  $T$  such that every leaf node of  $T$  has exactly one member of  $C$  on a path between it and the root. As a selectional profile, a tree cut will have a non-zero probability associated with every node in  $C$ , and a zero probability for all other nodes in  $T$ . Figure 1 below has examples of two tree cuts.

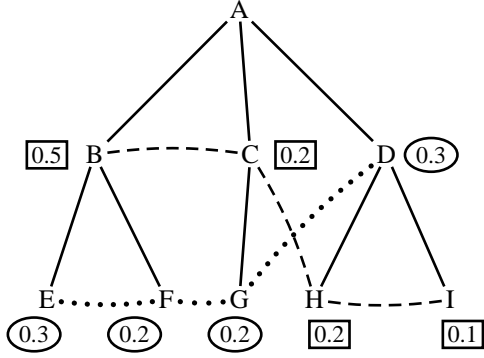


Figure 1: An example of two selectional profiles;  $profile_1$  in square boxes, and  $profile_2$  in ovals. Probability values of zero are not shown.

KL divergence, proposed by Lee, 2001) as a probability distance measure. The value of the distance measure was compared to a threshold, which determined classification of a verb as causative (the two profiles were similar) or non-causative (the two profiles were dissimilar), leading to best performance of 73% accuracy.

In McCarthy (2000), an error analysis reveals that the best method has more false positives than false negatives—some slots are considered overly similar because the selectional profiles are compared at a coarse-grained level, losing fine semantic distinctions.

In the next section, we propose an alternative method of comparing selectional profiles, which addresses the problem of insufficient discrimination of profile similarity in WordNet. Furthermore, the approach applies generally to any probability distribution over WordNet, unlike McCarthy’s method which is specific to profiles that are tree cuts.

### 3 Selectional Profile Distance

Our measure of selectional profile distance (SPD) is designed to meet two criteria. First, it should allow easy comparison between selectional profiles as probability scores spread throughout a hierarchical ontology (such as WordNet), not just between tree cuts. Second, it should capture fine-grained semantic similarity between profiles. To achieve these two goals, we measure the distance as a tree distance between the two profiles within the hierarchy, weighted by the probability scores.

(Note that we formulate a *distance* measure, while referring to a component of semantic *similarity*. We assume throughout the paper that WordNet distance is the inverse of WordNet similarity, and indeed the similarity measures we use are directly invertible.)

We illustrate with an example the differences between our measure and both McCarthy’s (2000) method and general vector distance measures. Consider the two selec-

tional profiles in Figure 1, with  $profile_1$  in square boxes, and  $profile_2$  in ovals.<sup>4</sup> To calculate the vector distance between  $profile_1$  and  $profile_2$ , we need two vectors of equal dimension. In this example, one can propagate the distributions to the lowest common subsumers (i.e., B, C, and D) as in McCarthy (2000). The vectors representing the two profiles become:

$$profile_1 = [0.5, 0.2, 0.3]$$

$$profile_2 = [0.5, 0.2, 0.3]$$

Alternately, one can also increase the dimension of each profile to include all nodes in the hierarchy (or just the union of the profile nodes). The two profiles become:

$$profile_1 = [0, 0.5, 0.2, 0, 0, 0, 0, 0.2, 0.1]$$

$$profile_2 = [0, 0, 0, 0.3, 0.3, 0.2, 0.2, 0, 0]$$

In the first method (that of McCarthy, 2000), the two profiles become identical. By generalizing the profiles to the lowest common subsumers, we lose information about the semantic specificity of the profile nodes and can no longer distinguish the semantic distance between the nodes across profiles. In the second method, the information about the hierarchical structure (of WordNet) is lost by treating each profile as a vector of nodes. Hence, vector distance measures fail to capture any semantic similarity across different nodes (e.g., the value of node B in  $profile_1$  is not directly compared to the value of its child nodes E and F in  $profile_2$ ).

To remedy such shortcomings, our goal is to design a new distance measure that (i) compares the *distributional* differences between two profiles (somewhat similar to existing vector distances), and also (ii) captures the *semantic* distance between profiles. Intuitively, we can think of the profile distance as how far one profile (source) needs to “travel” to reach the other profile (destination). Formally, we define SPD as:

$$SPD(profile_{src}, profile_{dest}) = \sum_{\substack{s \in profile_{src} \\ d \in profile_{dest}}} amount(s, d) * distance(s, d) \quad (1)$$

where  $amount(s, d)$  is the portion of the profile score at node  $s$  in  $profile_{src}$  that travels to node  $d$  in  $profile_{dest}$ , and  $distance(s, d)$  is the semantic distance between node  $s$  and node  $d$  in the hierarchy. For now, it can be assumed that  $amount(s, d)$  is  $score(s)$ , the entire probability score at node  $s$ . Note that we design the distance to be symmetric, so that the distance remains the same regardless of which profile is source and which is destination. (We present our distance measures below.)

<sup>4</sup>Note that these are both tree cuts, so that we can compare McCarthy’s method, but keep in mind that our method—as well as traditional vector distances—will apply to any probability distribution over a tree.

In the current example, we can propagate *profile<sub>2</sub>* (source) to *profile<sub>1</sub>* (destination) by moving its probabilities in this manner:

1. probabilities at nodes E and F to node B
2. probability at node G to node C
3. probability at node D to nodes H and I

The first two steps are straightforward—whenever there is one destination node in a propagation path, we simply multiply the amount moved by the distance of the path ( $distance(s, d)$ ). For example, step 1 yields a contribution to  $SPD(profile_{src}, profile_{dest})$  of  $score(E)dist(E, B) + score(F)dist(F, B)$ .

However, the last step, step 3, has multiple destination nodes (H and I), and the probability of the source node, D, must be appropriately apportioned between them. We take this into account in the *amount* function, by including a weight component:

$$amount(s, d) = weight(d) * portion(s) \quad (2)$$

where  $weight(d)$  is the weight of the destination node  $d$  and  $portion(s)$  is the portion of  $score(s)$  that we are moving. (For this example, we continue to assume that the full amount of  $score(s)$  is moved; we discuss  $portion(s)$  further below.) The weight of each destination node  $d$  is calculated as the proportion of its score in the sum of the scores of its siblings. Thus, in step 1 above,  $weight(B)$  and  $weight(C)$  are both 1, and the full amount of E, F, and G are moved up. In the last step, however, the sibling nodes H and I have to split the input from node D: node H has weight  $score(H)/(score(H) + score(I)) = 0.2/(0.2 + 0.1) = 2/3$ , and node I analogously has weight  $1/3$ .<sup>5</sup>

Hence, the SPD propagating from *profile<sub>2</sub>* to *profile<sub>1</sub>* can be calculated as:

$$\begin{aligned} SPD(profile_2, profile_1) &= score(E)dist(E, B) \\ &+ score(F)dist(F, B) + score(G)dist(G, C) \\ &+ \frac{2}{3}score(D)dist(D, H) \\ &+ \frac{1}{3}score(D)dist(D, I) \end{aligned}$$

For simplicity, we designed this example such that the two profiles are very similar. As a result, we end up propagating the *entire* source profile by propagating the

<sup>5</sup>We have described the algorithm as moving one profile to another. Conceptually, there are cases, as illustrated in the example, where we are propagating profile scores downwards in the hierarchy. Moving scores downwards can be computationally expensive because one may need to search through the whole subtree rooted at the source node for destination nodes. We implemented an alternative by moving all the scores upwards. Since we keep track of the source and destination nodes, the two methods are equivalent.

full score of each of its nodes. In practice, for most profile comparisons, we only move the portion of the score at each node necessary to make one profile resemble the other. Hence,  $portion(s)$  in the formula for  $amount(s, d)$  in equation 2 captures the difference between probabilities at node  $s$  across the source and destination profiles.

So far we have discussed very little the calculation of semantic distance between profile nodes (i.e.,  $distance(s, d)$  in equation 1). Recall that one important goal in designing SPD is to capture semantic similarity between WordNet nodes. Naturally, we look to the current research comparing semantic similarity between word senses (e.g., Budanitsky and Hirst, 2001; Lin, 1998). We choose to implement two straightforward methods. For one, we invert (to obtain distance) the WordNet similarity measure of Wu and Palmer (1994), yielding:

$$d_{wp}(n_1, n_2) = \frac{depth(n_1) + depth(n_2)}{2depth(LCS(n_1, n_2))}, \quad (3)$$

where  $LCS(n_1, n_2)$  is the lowest common subsumer of  $n_1$  and  $n_2$ . The other method we use is the simple edge distance between nodes,  $d_{edge}$ .<sup>6</sup>

Thus far, we have defined SPD as a sum of propagated profile scores multiplied by the distance “travelled” (equation 1). We have also considered propagating other values as a function of profile scores. Let’s return to the same example but redistribute some of the probability mass of *profile<sub>2</sub>*: node E goes from a probability of 0.3 to 0.45, and node F goes from 0.2 to 0.05. As a result, the distribution of the scores at the node B subtree is more skewed towards node E than in the original *profile<sub>2</sub>*.

For both the original and modified *profile<sub>2</sub>*, SPD has the same value because we are moving a total probability mass of 0.5 from E and F to B, with the same semantic distance (since E and F are at the same level in the tree). However, we consider that, at the node B subtree, *profile<sub>1</sub>* is less similar to the skewed *profile<sub>2</sub>* than to the original, more evenly distributed *profile<sub>2</sub>*. To reflect this observation, we can propagate the “inverse entropy” in order to capture how evenly distributed the probabilities are in a subtree. We define an alternative version of  $amount(s, d)$  as:

$$amount_e(s, d) = weight(d) * entropy_{inv}(s) \quad (4)$$

where we replace  $portion(s)$  with inverse entropy,  $entropy_{inv}(s)$ , which we define as:

$$entropy_{inv}(s) = \frac{1}{portion(s) \log_2 portion(s)} \quad (5)$$

<sup>6</sup>We also implemented the WordNet edge distance measure of Leacock and Chodorow (1998). Since it did not influence our results, we omit discussion of it here.

By propagating inverse entropy, we penalize cases where the distribution of source scores is “skewed.” In this work, we will experiment with both methods of propagation (with and without inverse entropy).

## 4 Materials and Methods

### 4.1 Corpus Data

Our materials are drawn from a 6M-word corpus of medical texts, which we mined for a related project. The texts are medical journal abstracts and articles obtained by querying the PubMed Central search engine (<http://www.pubmedcentral.nih.gov/>). Query terms were taken from entries listed under the “Medical Encyclopedia” and “Drug Information” sections of the MedlinePlus website (<http://www.nlm.nih.gov/medlineplus/>). The text is parsed using the RASP parser (Briscoe and Carroll, 2002), and subcategorizations are extracted using the system of Briscoe and Carroll (1997). The subcategorization frame entry of each verb includes the frequency count and a list of argument heads per slot. The target slots in this work are the subject of the intransitive and the object of the transitive.

### 4.2 Verb Selection

We evaluate our method on the causative alternation in order for comparison to the earlier methods of McCarthy (2000) and Merlo and Stevenson (2001). We selected target verbs by choosing *classes* (not individual verbs) from Levin (1993) that are expected to undergo the causative alternation. We refer to these as causative verbs. For our first development set, we chose filler (non-alternating) verbs from a small set of classes that are not expected to exhibit the causative alternation. These are the restricted-class verbs. For our second development set, we did not restrict the classes of the fillers, except to avoid classes that allow a subject/object alternation as in the causative. These are the broad-class verbs.

(Note that we did not hand-verify that individual verbs allowed or disallowed the alternation, as McCarthy (2000) had done, because we wanted to evaluate our method in the presence of noise of this kind.)

Verbs that occur a minimum of 10 times per frame are chosen. We randomly select 36 causative verbs and 36 filler verbs for development, forming two sets of 18 causative and 18 filler verbs. The first development set uses 18 restricted-class filler verbs, and the second uses 18 broad-class filler verbs. We also randomly select 20 causative verbs and 20 broad-class verbs for testing. (The 20 filler test verbs are all drawn from the same classes as the broad-class development verbs, so that we could directly compare performance between the second development set and the test set.)

Each set of verbs is further divided into a high frequency band (with at least 90 instances of one target slot), and a low frequency band (with between 20 and 80 instances of one target slot). These bands have 10 and 8 verbs, respectively, in the development sets, and equal numbers of verbs (10 each) in the test set. For each of the development and testing phases, we experiment with individual frequency bands (i.e., high band and low band, separately), and with mixed frequencies (i.e., all verbs).

### 4.3 Experimental Set-Up

For each verb, we extracted the argument heads of the target slots from the corpus. Using (verb,slot,noun) frequencies, we experimented with several ways of building selectional profiles of each verb’s argument slot (Resnik, 1993; Li and Abe, 1998; Clark and Weir, 2002).<sup>7</sup> In our development work, we found that the method of Clark and Weir (2002) overall gave better performance, and so we limit our discussion here to the results on their model. It is worth noting that the method of Clark and Weir (2002) does not yield a tree cut, but instead generally populates the WordNet hierarchy with non-zero probabilities. This means that the kind of straightforward propagation method used by McCarthy (2000) is not applicable to selectional profiles of this type.

We compare SPD to a number of other measures, applied directly to the (unpropagated) probability profiles given by the Clark-Weir method: the probability distribution distances given by skew divergence (skew) and Jensen-Shannon divergence (JS) (Lee, 2001), as well as the general vector distances of cosine (cos), Manhattan distance (L1 norm), and euclidean distance (L2 norm).

To determine whether a verb participates in the causative alternation, we adopt McCarthy’s method of using a threshold over the calculated distance measures, testing both the mean and median distances as possible thresholds. In our case, verbs with slot-distances below the threshold (smaller distances) are classified as causative, and those above the threshold as non-causative. Accuracy is used as the performance measure.

## 5 Experimental Evaluation

We evaluate the SPD method on selectional profiles created using the method of Clark and Weir (2002), with comparison to the other distance measures as explained above. In the calculation of SPD, we compare the two node distance measures,  $d_{wp}$  (Wu and Palmer, 1994) and  $d_{edge}$ , and the two ways of propagating selectional profiles, without entropy ( $e0$ ) and with entropy ( $e1$ ), as de-

<sup>7</sup>Recall that a selectional profile is a probability distribution over WordNet. Although Resnik’s measure is not a probability distribution, his method for populating the WordNet hierarchy from corpus counts does yield a probability distribution.

Dev 1 (with Restricted Class Fillers)					
Average Threshold			Median Threshold		
all	high	low	all	high	low
0.64	0.65	0.62	0.67	0.7	0.75
SPD	SPD	cos	SPD	SPD	cos
cos					
Dev 2 (with Broad Class Fillers)					
Average Threshold			Median Threshold		
all	high	low	all	high	low
0.69	0.65	0.75	0.67	0.7	0.75
SPD	SPD	SPD	SPD	L1	SPD
	L1	cos		L2	cos
	skew			skew	
	JS				

Table 1: The best accuracy achieved in each condition (development set and threshold), along with the measure(s) that produce that result. SPD refers to SPD without entropy, using either  $d_{wp}$  or  $d_{edge}$ . “all”, “high”, and “low” refer to the different frequency bands.

scribed in Section 3. These settings are mentioned when relevant to distinguishing the results.

### 5.1 Development Results

On the two development sets, SPD generally performs better than the other measures. In particular, our measure achieves a best accuracy of 69% (random baseline of 50%, broad class fillers, all verbs). The best performance is compiled in Table 1. Observe that in each condition, SPD (without entropy, using either  $d_{wp}$  or  $d_{edge}$ ) is always the best (or tied for best) at classifying all verbs, and at classifying at least one other frequency band. No other measure performs consistently as well as SPD. Indeed, on closer examination, in the cases where SPD is not the best, it has the second best performance. Interestingly, we also discover that cosine works well in the low frequency band.

There is only a small difference in the SPD performance between the two development sets. Recall that broad class fillers contain non-causatives from a wider variety of classes than restricted class fillers, which we thought would make the classification task harder, because of more variation in the data. However, not only is the broad class performance not lower, there are some cases in which it surpasses the restricted class performance. At least for these verbs, amount of variation in the classes has little impact.

SPD with entropy does not perform best on development verbs. However, in comparison to the vector distance measures (which yield below chance accuracies in most cases), SPD with entropy does achieve reasonable accuracies. It is always above chance, and sometimes second best.

Generally, across both development sets, using a me-

Unseen Test Verbs			
	all	high	low
Best	0.65	0.7	0.8
	SPD <sub>e1</sub>	cos	SPD <sub>e1</sub>
	cos		
2nd Best	0.6	0.6	0.7
	SPD <sub>e0</sub>	SPD <sub>e0</sub>	cos
		SPD <sub>e1</sub>	skew

Table 2: The best and second best accuracy achieved in testing, along with the measure(s) that produced the result, using a median threshold. “all”, “high”, and “low” refer to the different frequency bands.

dian threshold works somewhat better than an average threshold. To focus our testing phase, we use only the median threshold.

### 5.2 Test Results

Table 2 shows both the best and second best results in the testing phase. Here, similarly to the development results, SPD is the best (or tied for best) at classifying all verbs, and verbs in the low frequency band. In cases where it is not the best, it is the second best.

Contrary to the development results, SPD measures with entropy, SPD<sub>e1</sub>, fare somewhat better than those without entropy, SPD<sub>e0</sub>. To examine the difference in performance, we do a pairwise comparison of the actual verb classification. In the “all” frequency case, SPD with entropy has 7 false positives,<sup>8</sup> and SPD without entropy has 8 false positives, 5 of which are misclassified by both. Furthermore, with the exception of one verb, the remaining false positives are quite near the threshold. The trends in the low frequency band are quite similar—there is considerable overlap between SPD<sub>e0</sub> and SPD<sub>e1</sub> false positives. Given the similarity of the classifications, we conclude that the propagation methods (with or without entropy) would likely be comparable on larger sets of verbs.

Recall that we also experiment with two different node distance measures ( $d_{wp}$  and  $d_{edge}$ ). Interestingly, the performance between the two is remarkably similar. In fact, the actual classifications themselves are very similar. Note that Wu and Palmer (1994) designed their measure such that shallow nodes are less similar than nodes that are deeper in the WordNet hierarchy. This property is certainly lacking in the edge distance measure. Here we can only speculate that perhaps our selectional profiles are relatively similar in terms of depth, so that taking relative depth into account in the distance measure has little impact.

For comparison, we replicate McCarthy’s method,<sup>9</sup>

<sup>8</sup>Hence, 14 are misclassified, since we use median, which splits the verbs exactly in half into the two classes.

<sup>9</sup>We replicate McCarthy’s method using tree cuts produced

which only achieves above chance performance in a few cases: on the development verbs with restricted fillers (56%, low frequency verbs, average threshold), and on the development verbs with broad class fillers (58%, all verbs, average threshold; and 62%, low frequency verbs, median threshold). This result is very different from her reported results. One major difference between our experimental set-up and hers is the selection of verbs. We do not hand-select our causative verbs to ensure they undergo the causative alternation. We speculate that there is more noise in our data than in McCarthy’s and our method is less sensitive to that.

One puzzle in the pattern of results is the cosine performance—cosine has the best or second best accuracy across all bands in the test data, while it is best mostly in the low band in development. We are a bit surprised that cosine works well at all. In the future, we intend to examine the conditions where cosine is a sufficient discriminator.

### 5.3 Frequency Bands

Somewhat surprisingly, we often get better performance with both the low and high frequency bands individually than we do with all verbs together. By inspection, we observe that low frequency verbs tend to have smaller distances between two slots and high frequency verbs tend to have larger distances. As a result, the threshold for all verbs is in between the thresholds for each of the frequency bands. When classifying both types of verbs, the frequency effect may result in more false positives for low frequency verbs, and more false negatives for high frequency verbs.

We examine the combined performance of the individual frequency bands, in comparison to the performance on all verbs. Here, we define “combined performance” as the average of the accuracies from each frequency band. (The averages are weighted averages if each band contains a different number of verbs.) We find that  $SPD_{e1}$  attains an averaged accuracy of 70%, an improvement of 5% over the best accuracy classifying all verbs together. Separating the frequency bands is an effective way to remove the frequency effect.<sup>10</sup>

Stemming from this analysis, a possible refinement to separating the frequency bands is to use a different classifier in each frequency band, then combine their performance. We observe that combining the best accuracies gives us an accuracy of 75% (best low band accuracy of 80% and best high band accuracy of 70%), outperform-

---

by Li and Abe’s technique, which are propagated to their lowest common subsumers and their distance measured by skew divergence.

<sup>10</sup>Another method is to use some type of “expected distance” as a normalizing factor (Paola Merlo, p.c.). However, it is yet unclear how we would calculate this number.

ing the “all verbs” best accuracy by 10%. Although in our current results there is no one classifier that is clearly the best overall for a particular frequency band, we plan to examine further the relationship between verb frequency and various distance measures.

## 6 Conclusions

We have proposed a new method for comparing WordNet probability distributions, which we call selectional profile distance (SPD). Given any pair of probability distributions over WordNet (which we call a selectional profile), SPD captures in a single measure the aggregate semantic distance of the component nodes, weighted by their probability. The method addresses conceptual problems of an earlier measure proposed by McCarthy (2000), which was limited to tree cut models (Li and Abe, 1998) and failed to distinguish detailed semantic differences between them. Our approach is more general, since it can work on the result of any model that populates WordNet with probability scores. Moreover, the integration of a WordNet distance measure into the formula enables it to take semantic distances directly into account and better capture meaningful distinctions between the distributions.

We have shown that SPD yields practical advantages as well, in demonstrating improved performance in the ability to detect a verb alternation through comparison of the selectional profiles of potentially alternating slots. SPD achieves a best performance of 70% accuracy (baseline 50%) on unseen test verbs, and no other measure we tested performed consistently as well as it did, achieving best performance (alone or tied) in 9 of 12 development experiments, and best or second best in all three test scenarios. By comparison, McCarthy (2000) attained 73% accuracy on her set of hand-selected test verbs in a similar task; however, when applied to our various sets of randomly selected verbs, our replication of her method performed very poorly, rarely reaching above chance performance. We believe that the randomly selected verbs in our experiments may show a wider variation, than verbs that are hand-selected, in whether and how much they alternate, and thus constitute a more difficult but more realistic scenario for testing the usefulness of these measures in practice.

Interestingly, we found that separating verbs into low and high frequency bands improved performance, and our best performance of 70% in fact results from an average of SPD results on the individual frequency bands. Perhaps even more interesting is the underlying reason for this: causative verbs in the low frequency band show greater similarity (lower SPD scores) across the slots than those in the high frequency band. In on-going work, we are extending our experiments to a larger corpus (the BNC), so that we can investigate a larger range and num-

ber of verbs to explore this issue, which will enable us to better elucidate the reasons for this interaction.

## 7 Acknowledgments

We thank Diana McCarthy from University of Sussex for providing the tree cut acquisition code, David James from University of Toronto for pre-processing the corpus data, and Ali Shokoufandeh from Drexel University and Ted Pedersen from University of Minnesota for helpful discussion. We gratefully acknowledge the support of NSERC and OGS of Canada.

## References

- J. Allen. 1997. Probabilistic constraints in acquisition. In *Language Acquisition: Knowledge Representation and Processing*, Edinburgh, UK.
- C. Bannard. 2002. Statistical techniques for automatically inferring the semantics of verb-particle constructions. Master's thesis, University of Edinburgh, Edinburgh, UK.
- T. Briscoe and J. Carroll. 1997. Automatic extraction of subcategorization from corpora. In *Proceedings of the 5th Applied Natural Language Processing Conference*, p. 356–363, Washington, D.C.
- T. Briscoe and J. Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, p. 1499–1504, Las Palmas, Canary Islands.
- A. Budanitsky and G. Hirst. 2001. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources*, p. 29–34.
- S. Clark and D. Weir. 2002. Class-based probability estimation using a semantic hierarchy. *Computational Linguistics*, 28(2):187–206.
- H. T. Dang, K. Kipper, and M. Palmer. 2000. Integrating compositional semantics into a verb lexicon. In *Proceedings of the Eighteenth International Conference on Computational Linguistics*, Saarbrücken, Germany.
- B. J. Dorr and D. Jones. 2000. Acquisition of semantic lexicons: Using word sense disambiguation to improve precision. In E. Viegas, editor, *Breadth and Depth of Semantic Lexicons*, p. 79–98. Kluwer Academic Publishers, Norwell, MA.
- B. Katz, J. Lin, and S. Felshin. 2001. Gathering knowledge for a question answering system from heterogeneous information sources. In *Proceedings of the Workshop on Human Language Technology and Knowledge Management*, Toulouse, France.
- M. Lapata. 1999. Acquiring lexical generalizations from corpora: A case study for diathesis alternations. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, p. 397–404.
- M. Lapata and C. Brew. 1999. Using subcategorization to resolve verb class ambiguity. In *Proceedings of Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, p. 266–274, College Park, MD.
- C. Leacock and M. Chodorow. 1998. Combining local context and WordNet similarity for word sense identification. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, p. 265–283. MIT Press.
- L. Lee. 2001. On the effectiveness of the skew divergence for statistical language analysis. In *Artificial Intelligence and Statistics*, p. 65–72.
- B. Levin. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press.
- H. Li and N. Abe. 1998. Generalizing case frames using a thesaurus and the MDL principle. *Computational Linguistics*, 24(2):217–244.
- D. Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning*, Madison, Wisconsin.
- D. McCarthy. 2000. Using semantic preferences to identify verbal participation in role switching alternations. In *Proceedings of Applied Natural Language Processing and North American Chapter of the Association for Computational Linguistics (ANLP-NAACL 2000)*, p. 256–263, Seattle, WA.
- P. Merlo and S. Stevenson. 2001. Automatic verb classification based on statistical distributions of argument structure. *Computational Linguistics*, 27(3):393–408.
- P. Resnik. 1993. *Selection and Information: A Class-Based Approach to Lexical Relationships*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.
- S. Schulte im Walde and C. Brew. 2002. Inducing German semantic verb classes from purely syntactic subcategorisation information. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA.
- S. Teufel. 1999. *Argumentative Zoning: Information Extraction from Scientific Articles*. Ph.D. thesis, University of Edinburgh, Edinburgh, UK.
- V. Tsang, S. Stevenson, and P. Merlo. 2002. Crosslinguistic transfer in automatic verb classification. In *Proceedings of the 19th International Conference on Computational Linguistics*, Taipei, Taiwan.
- Z. Wu and M. Palmer. 1994. Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, p. 133–138, Las Cruces, New Mexico.