Lexical Acquisition of Verb Direct-Object Selectional Preferences Based on the WordNet Hierarchy

Emily Shen and Sushant Prakash
June 7, 2006

Introduction

Lexical acquistion is an important part of NLP, not only because it helps automatically add to outdated or insufficient man-made lexical resources, but also because it strives to find semantic information. Selectional preferences, in which some part of a sentence imposes semantic constraints on another part, is clearly important in language, particularly when looking at the relationship between verbs and their direct objects. For example, when one sees "eat", without knowing the subject or anything else in the sentence, one could say that the direct object is most likely some type of food. Similarly, the verb "drive" would most likely be followed by some type of vehicle. Switch those preferences around - eating a truck or driving an eggplant - and we get an unnatural, almost jarring (similar to reading a sentence with a grammar mistake) construction. Thus, being able to model the relationship between verbs and their direct object would be a valuable step in NLP research, possibly helping with areas such as parse decisions and the semantic inference of unknown words.

Resnik (1995) attempted to model V-DO (verb, direct-object) selectional preferences by having each noun seen contribute count mass to all of its classes as defined by WordNet. He then defined selectional strength of a verb and association strength between a verb and class based on the KL divergence between the the probability of the class, and the probablity of the class given the word. Li and Abe (1996) tried a different method, designing a learning algorithm based on the Minimum Description Length principle to cluster a cartesian product of a set of verbs and a set of nouns. McCarthy (1997) built upon Li and Abe's work by coupling the system with a Word Sense Disambiguator.

Problem

We attempt to improve on Resnik's model, which cannot deal with nouns not seen in training and does not take into account the hierarchical structure of WordNet, by acquiring direct-object preferences for various verbs using statistics obtained from corpora and the semantic information contained within WordNet. In particular, we investigate the benefit of using the hierarchical structure to transfer information between nouns. The method presented in the textbook and by Resnik assumes a flat class structure of nouns. Extending the method to the hierarchical structure to obtain better information is a nontrivial problem since there are different ways to use the structural information.

Approach

We first build a model using a flat class procedure similar to Resnik's. A class is considered to be a synset, Wordnet's node for a fundamental concept or meaning (a word may have several synsets, each corresponding to a sense of the word - e.g. "airline" has two synsets, one for a hose that carries air, and one for a flight company). We will use double quotes to refer to a word, as in a string of letters, and use single quotes to refer to a class, as in a synset or concept in

WordNet. With Resnik's model, each verb is assigned a selectional strength based on the KL divergence between the prior distribution of the objects and the distribution of the objects for the particular verb in question:

$$S(v) = D(P(C|v)||P(C)) = \Sigma_c P(c|v)\log[P(c|v)/P(c)]$$

S(v) is then used to calculate the association strength between a verb and a class by taking the proportion that that class contributes to the sum:

$$A(v,c) = P(c|v)\log[P(c|v)/P(c)] / S(v)$$

Here, $P(c|v) = P(v,c)/P(v)$, which are given the maximum likelihood estimate:

$$P(v) = C(v) / \Sigma_{v'}C(v')$$

$$P(v,c) = (1/N)\ \Sigma_{n\in words(c)}(1/|classes(n)|)\ C(v,n)$$

where N is the number of V-DO pairs seen. With this estimate of P(v,c), we distribute the count for a particular verb noun pair uniformly across all the senses that n has, since in our data we are given no information as to what sense a noun refers to.

The association strength between a verb and a noun is the maximum association strength the verb has with a class that the noun is a part of:

$$A(v,n) = \max_{c\in classes(n)} A(v,c)$$

The advantage of defining our strengths this way is that in addition to the associations between particular verbs and nouns, we also get a measure of how "general" a verb is - that is, a high S(v) tells us the verb is very selective in the direct objects it takes, while a low S(v) indicates the verb will take a variety of classes as direct objects.

A problem with this flat class model can be immediately seen: synsets that are related to each other do not share information. For example, having seen <v,'grape'>, <v,'apple'>,<v,'banana'> for some verb v, one would guess that v takes fruits and that it would therefore have a high association with 'orange'. But in the flat class model, 'grape', 'apple', 'banana', and 'orange' are considered different classes with no relationship at all.

To deal with this defect, we built a modified model which, when calculating the associations for a verb, first sets P(v,c) for classes the same as the flat model, but then propagates each of these counts up to k ancestors of c, where k can be any integer from 0 to infinity (infinity denoting that the counts are propagated all the way up to the root node 'entity'). Thus the modified probability for a verb v and class c is the original probability of v,c plus the original probabilities of all classes c_{kdes} that are descendents of c within k levels:

$$P_{mod}(v,c) = P_{orig}(v,c)+\Sigma_{c_{kdes}}P_{orig}(v,c_{kdes})$$

The rationale behind this approach is that if an instance of v,c is seen, than we can assume we have seen an instance of v,c_{ans} where c_{ans} is a hypernym of c. For example, if we see <'hunt', "deer">, we can treat "deer" as an instance of the class 'deer' or as an instance of the class 'animal'. While it may seem reasonable to distribute the count of <"hunt","deer"> among the classes 'deer' and 'animal', we feel that this would be the wrong approach since the classes are not disjoint concepts: that is, "deer" is not referring to something that is either a 'deer' or an 'animal' (in the way that an instance of a word only refers to one particular sense of that word), but rather "deer" is both a 'deer' and an 'animal' at the same time.

As it happens, a synset in WordNet can have more than one direct hypernym. For example 'milk' (referring to a white nutritious liquid secreted by mammals and used as food by humans) has two parent classes: 'dairy product' and 'beverage'. Even in this case, the probabilities were propagated in the same manner, without splitting it between all parents. The same rational as above holds for this decision: we do not have to choose between two classes, since 'milk' can be considered to be both a 'dairy product' and a 'beverage'. Care was taken, however, to avoid double-counting probabilities when propagating: when the hypernym paths of 'dairy product' and 'beverage' merge at 'food, nutrient', the probability for 'milk' would only be added once to it.

Besides splitting the probability among the hypernym classes, discounting the probability as one moves farther and farther away from the original class may seem like the right thing to do, the idea being that the higher class c is from the original class c_{orig} the less likely it means that seeing <v,c_{orig}> could also mean seeing <v,c>. However, it must be taken into account that what we ultimately compute is A(v,c) which involves the KL divergence between P(C) and P(C|v). When P(c|v) is calculated under the modified model, it can be interpreted as "What is the probability that verb v will take class c or any of c's descendants as direct objects?" and P(c) can be interpreted as "What is the probability that a direct object is class c or any of c's decendants". As we move up the hypernym tree, the classes become more and more general and thus P(c) and P(c|v) become more and more similar, resulting in lower A(v,c). At the extreme end, for c='entity', the root node for all nodes in WordNet, P(c)=P(c|v)=1 and thus A(v,c)=0 for any verb v. Thus, since this discount is already inherent in the calculation of the association strengths, we do not need to account for it when propagating. The strategy behind our approach is to find the node(s) that characterize the most general sorts of objects that a verb can take - on one hand, general nodes are favored because they get probabilities from all of their descendants, but on the other hand, P(c) also rises with generality, bringing down A(v,c). With these two opposing forces, the idea is that the model will settle on some intermediate node -thus our model would not choose 'corn' (too specific) or 'object' (too vague) for the class most associated with "eat", but would rather choose 'food' instead.

To calculate A(v,n) goes through each synset i of n, and finds the max A(v,c) where c is an ancestor of synset i. We then return the max of these association strengths, taking note of which synset generated that max. In this way, our model chooses a sense of the noun that is most closely related to the verb (a result which we use to test the model).

Linguistic Assumptions

The concept of selectional preferences assumes that relationships such as verb-direct object are influenced by semantic content. This allows us to generalize from specific nouns as direct objects of a verb to classes of nouns. Our model considers only verb-direct object relationships, but it could be generalized to acquire selectional preferences for any kind of grammatical relationship within a sentence that can be extracted fairly accurately from part-of-speech tagging or parse trees. Our model does not take into account possible influences on the direct object of a verb from the subject. For example, "this car eats gas" is valid, whereas "the boy eats gas" is not (hopefully). Although one application of our verb-direct object selectional preference model is word sense disambiguation of objects, our model does not distinguish between multiple senses of verbs. The underlying assumption is that verbs determine direct objects much more strongly than direct objects determine verbs.

Data Collection

We obtained our data from the 1987 BLLIP corpus (~10 million words) and the WSJ corpus (~1 million words). The first step was to compile a list of verb - direct object pairs seen in our corpora. Since corpora are not usually tagged with such information, different ways of pursuing this step could lead to different purities of correct V-DO relations, and hence have a significant effect on the performance of our model. We pursued two methods: miniparring and tgrepping. For the former, we ran minipar on a collection of sentences, printing out the dependency parse of each one. We decided to keep only triples of the form "V:obj:N" even though there were occurences of triples with "V:obj1:N" and "V:obj2:N" for verbs with a direct and an indirect object. In that case the number for obj does not determine whether the object is a direct or indirect one, so we threw them out to keep as high a purity as possible. Since minipar only requires sentences rather than trees as input, its main attraction was that it could be run on very large corpora to get as much data as we wanted. However, since we noticed that the parses were not always correct, we decided to pursue the second method involving the use of hand-parsed corpora. To extract V-DO pairs, we used the following pattern to match nodes in the trees with tgrep2:

```
' /^VP/ <1 `/^VB/ <2 (/^NP/ <<= (/^NP/ !<< /^NP/ << (`/^NN/ !$. /^NN/)))'
```

This looks for a verb phrase where the first child is a verb and the second child is a noun phrase. In the noun phrase it finds all the lowest noun phrases and in each one of those looks at the rightmost noun. While the amount of data in tree form is significantly less, we thought the higher accuracy of human-parsed structures would make up for it.

The resulting pairs then went through another filter that removed pairs in which either of the words had one of the following properties:

1) not in Wordnet
2) only one letter
3) in the stop list :'an', 'as', 'at', 'by', 'he', 'his', 'me', 'or', 'thou', 'us', 'who', 'it', 'more', 'ii', 'iii', 'one', 'none', 'be', 'become'

Number of pairs obtained for each corpus:
~550k for 1987 BLLIP
~50k for WSJ miniparred
~45k for WSJ tgrepped

Testing

To test our model, we evaluated its success in word sense disambiguation of direct objects of verbs. Another possible method of evaluation would have been to output probabilities of direct object nouns given verbs and compare performance as a language model with a simple bigram model. However, our method of computing conditional probabilities $P(c|v)$ in the WordNet hierarchy does not lend itself well to computing probabilities of individual nouns $P(n|v)$. We could write $P(n|v) = P(c|v) P(n|c)$, but in order to compute meaningful probabilities of nouns given classes, we would actually want $P(n|c)$ to be the probability of a noun n given that class c is some ancestor (not necessarily a synset or immediate hypernym) of the noun. Also, it is not clear which class c would be used to calculate $P(c|v)P(n|c)$. We could calculate the value for each inherited hypernym of n and take the maximum value, but this would not result in a valid probability distribution for $P(n|v)$. Furthermore, given the goal of selectional preference acquisition, it seems that we should be evaluating the model on associations of verbs and classes of nouns rather than individual nouns. Finally, word sense disambiguation is particularly interesting to test on because the assumption used in computing initial leaf probabilities in the model is that all senses of a given word are equally likely, but with all of the training data, the model is able to tell that for a given verb certain senses of a direct object are more likely or less likely.

For the word sense disambiguation task, we made a test file of 80 verb object pairs, 65 of which had been seen at least once, and 15 of which were unseen in WSJ, along with a list of the senses of the noun we determined were possible given the verb. We hand picked these verb object pairs as ones for which we ourselves could disambiguate the (ambiguous) noun given only the verb. Specifically, this meant that given the verb, among the various WordNet senses of the noun, we were able to distinguish the sense (or a few similar senses) that could definitely occur with the verb, and the remaining senses were ones that would almost definitely not occur with the verb. For example, two of our test pairs were ("make", "promise") and ("show", "promise"). For the verb "make", "promise" can be disambiguated to the sense 'a verbal commitment by one person to another...', and for the verb "show", "promise" can be disambiguated to the sense 'grounds for feeling hopeful about the future'. Another test pair was ("drink", "juice") since "juice" can be disambiguated to 'beverage' and not 'energetic vitality', 'electric current', or 'any of several liquids of the body'.

For each verb object pair in the test file, the model ranked the various sense of the object given the verb in order by association strength. Intuitively, we wanted our model to give the correct sense or senses the highest association strength and rank the incorrect senses lower. We wanted to evaluate the model on precision, recall, and F1 (the harmonic mean of precision of recall). Letting P be the set of correct senses from our "gold standard" and A be the set of top ranked senses from our model, these can be calculated as follows:

Precision: $|A \cap P| / |A|$
Recall: $|A \cap P| / |P|$
F1: $2^* |A \cap P|/(|P|+|A|)$

However, the fact that the model is ranking senses rather than choosing a subset complicates things. Also, we do not want to require the model to give the exact same association strength to all sense in our "correct" set, but we want them to appear before the "incorrect" senses.

Therefore, we devised a few different ways of choosing the set A. We call this methods MaxPri, MaxGap, and AvgGap.

The first method, MaxPri (for maximum priority in a priority queue), chooses only the senses that are tied for the highest association strength. In order to allow senses that are close but not equal to the highest association strength to be considered, we came up with two other methods, MaxGap and AvgGap.
MaxGap, first ranks the senses by association strengths with the highest first. Then it looks at the difference in association strength between each pair of consecutive senses in the ranking and finds the maximum difference and separates the ranking into two around this difference. The top part becomes the set of correct senses A and the bottom part becomes the set of incorrect senses. For example, if the association strengths for senses 1, 2, 3, and 4 were 0.05, 0.04, 0.02, and 0.01, respectively, then the maximum difference would be 0.02, and model would choose senses 1 and 2 as correct senses of the noun. We came up with this second method because we want the model to split the senses into two groups based on the association strengths, and from looking at the output files we noticed that often there were clearly two groups of similar association strengths, one higher and one lower.

To account for cases where the difference between the highest association strength and the next highest is not quite the maximum difference but should be considered high enough to separate the first two senses, we devised the method AvgGap. Instead of separating based on the maximum difference as MaxGap does, AvgGap computes the average difference between consecutive association strengths (including zeros for ties), and then takes the top association strengths until the next difference is greater than some constant K times the average difference. For example if K = 1.5, and the association strengths for senses 1, 2, 3, and 4, were 0.01, 0.05, 0.07, and 0.13, respectively, then the average difference would be 0.02, and only sense 4 would be included in A since 0.13-0.07 = 0.06 > 1.5*0.02 = 0.03. We found empirically that K = 1.9 worked well.

The baseline that we compared to ranked all senses of a noun equally, giving low precision but 100% recall. We compared the performance of the models trained on WSJ minipar, WSJ tgrep2, and BLLIP 1987. For each of these three models, we compared the performance of our model that propagates leaf probabilities all the way through the hierarchy to a flat class structure. In addition, we tried just propagating probabilities some constant number of levels up in the hierarchy (1, 5, or 10). The idea behind this was that seeing a particular noun with a verb tells us something about nouns in similar classes but not necessarily about nouns in classes at very different levels in the tree.

Results

Here are the results of evaluating our models on the word sense disambiguation task.

		MaxPri			MaxGap			AvgGap		
		P	R	F1	P	R	F1	P	R	F1
Baseline		0.2639	1.0	0.4176	0.2639	1.0	0.4176	0.2639	1.0	0.4176
Flat	WSJ minipar	0.3104	0.7271	0.4351	0.3122	0.7771	0.4455	0.3103	0.7583	0.4404
	WSJ tgrep2	0.2839	0.7167	0.4067	0.2861	0.7479	0.4139	0.2853	0.7354	0.4111
	BLLIP 1987	0.3943	0.5833	0.4706	0.3828	0.7146	0.4986	0.3773	0.6583	0.4797

Hyper	WSJ minipar	0.4860	0.6854	0.5687	0.4053	0.7375	0.5231	0.4224	0.7188	0.5321
	WSJ tgrep2	0.5144	0.6521	0.5751	0.4387	0.7771	0.5608	0.4615	0.7333	0.5665
	BLLIP 1987	0.5176	0.6646	0.5820	0.4260	0.7333	0.5390	0.4720	0.7208	0.5705
WSJ tgrep2 prop1		0.4907	0.55	0.5186	0.4670	0.6417	0.5406	0.4564	0.6125	0.5230
WSJ tgrep2 prop5		0.5448	0.5896	0.5663	0.5098	0.6730	0.5801	0.5105	0.6354	0.5661
WSJ tgrep2 prop10		0.5060	0.6521	0.5698	0.4178	0.7708	0.5419	0.4503	0.7271	0.5560

Fig. 1. Precision, Recall, F1 scores for our models on WSD task.

For a fixed corpus, as we move from a flat class structure to propagating probabilities a few levels up to propagating probabilities all the way to 'entity', our precision generally increases, our recall decreases, and our F1 score increases. This indicates that propagating probabilities farther up through the hierarchy does help. We can see that WSJ tgrep2 outperforms WSJ minipar, indicating that the difference between hand parsing and automatic parsing makes a difference in our model. The three different strategies of computing precision and recall in fact gave fairly similar F1 results for each model. MaxGap and AvgGap give similar precision and recall, with precision being lower than for MaxPri and recall being higher than for MaxPri, and overall F1 being slightly less than for MaxPri. So in fact our original strategy of MaxPri did the best because of the loss in precision from the other methods. Our best F1 performance was with BLLIP 1987 propagating counts all the way up; this gave 52% precision, 66% recall, and 58% F1, compared to 26% precision, 100% recall, and 42% F1 for the baseline.

Other Evaluation

Another method we used to test our model was to rank the verbs v we had seen in order by their selection preference S(v) and check that the ranking matched with our intuition about how strongly each verb influences the class of nouns that appear as its object. Figure 2 shows the verbs with the highest and lowest selectional preferences (training on hyper BLLIP 1987).

Most selective verbs	Least selective verbs
discipline:112.62	have:2.26
slice:109.53	include:4.51
sigh:109.45	make:4.75
shoot down:104.86	see:4.96
elongate:100.86	get:5.25

Fig. 2 Most selective and least selective verbs and their selectional preferences for hyper BLLIP 1987.

The rankings match fairly well with our intuition. The verbs with the lowest selectional preferences ("get", "see", "make", "include", "have") seem to be among the least selective of direct object classes. Most of the verbs with the highest selectional preferences seem to be highly selective. The appearance of the verb "sigh" seemed possibly erroneous, since the verb does not usually take direct objects (except for "breath" as in "a breath of relief", for example). We looked into this and found that it was because our training data contained one object, "San Francisco", for "sigh", as a result of an incorrect dependency parse. To give some further intuition about highly selective verbs, "drink" is in the top 3% and "eat" in the top 30% of selective verbs out of the 2589 verbs which appeared in training.

We also output the top 5 and bottom 5 association strengths A(v,c) and the associated class for each verb v. These generally matched with our intuition as well. For example, for the verb "create", the top 5 classes were synsets of "state", "attribute", "condition", "group", and "social group", and the bottom 5 classes were synsets of three different senses of "life", "electric resistance", and "wrath". For the verb "stop", the top 5 classes were synsets of "event", "act", "psychological feature", "transaction", and "commerce". These last two make sense given that our souce is the WSJ. The bottom 5 classes were 'watercourse', 'meditation', 'coal', 'troy unit', and 'marketplace'. For the verb "transplant", the top 4 classes were 'excretory organ', 'kidney', 'internal organ', and 'body part', and for the verb "plant", the top 5 classes included 'plant' and 'explosive device', even though those actual words were never seen as objects of the verb.

Success and Error Analysis

On the test file, our model was able to disambiguate noun senses correctly a good proportion of the time. For example, our model correctly disambiguates "product" to the two mathematical over four other senses for the verb is "calculate", and disambiguates "slate" to 'a list of candidates nominated by a political party' over the three other senses that have to do with rock for the verb "propose". There are several examples that our BLLIP1987 model gets correct that WSJ miniparred does not, and this can be attributed to the larger training set. Looking at the examples our model gets wrong tells us about some of the shortcomings of the model. Here are a few of them.

Example 1:
("raise", "pet")
The senses of "pet" are 'a domesticated animal...', 'darling, favorite', 'a fit of petulance or sulkiness', and 'positron emission tomography'. We would expect our model to disambiguate "pet" correctly to the first sense, but in fact it chooses 'a fit of petulance' and 'positron emission tomography'. Upon examing the training pairs, we found that "raise" had been seen 6347 times (so there were plenty of training examples), with the most common objects being "question" (475), "price" (467), "stake" (454), and "rate" (405). Some of the other objects are "argument", "suspicion", "fear", and "concern". Since "raise" is seen so often with objects that have "attribute" or "abstraction" as an inherited hypernym (as 'a fit of petulance' and 'positron emission tomography' both do) and "a domesticated animal" is a physical entity, not an abstraction, our model chooses two incorrect senses over the correct sense. This example shows the shortcoming of not distinguishing between various senses of verbs. This problem clearly involves distinct senses of the verb "raise". However, given that our training data is not tagged with word senses, it would be difficult to take different verb senses into account. One approach would be to try to also define selectional preferences of noun direct objects for verb classes and use this to try to disambiguate the verbs. One difficulty we foresee with this approach is that it essentially defines some sort of clustering of the verbs into senses but does not provide a way to match the senses to specific WordNet senses of the verb. This problem would require further investigation.

Example 2:
("teach", "course")

There are 8 senses of the word "course", only one of which we considered correct: "course of study, course of instruction, class". Intuitively it seems that the verb "teach" is fairly selective and this disambiguation would not be hard for our model. Our model chose 4 incorrect senses over the correct sense. To see why this is the case, we looked at the frequency of direct objects for "teach" and found that of 326 occurrences, 20 were with "course" and 10 were with "class". Both "course" and "class" have many senses, so their count mass is distributed thinly among different senses. Since so much of the count mass for "teach" went to words that have only one relavent sense but many irrelevant ones, there was not enough count mass from other words to help disambiguate among the senses of the common nouns.

Future Work

One source of error was that the nouns in our training data were not disambiguated when we calculated P(v,c), forcing us to distribute uniformly among the senses. An improvement to our model would be to train on disambiguated training data. An alternative, since our model ranks the likelihood of senses of a noun given a verb, would be to feed the disambiguation results of the model back into the model in a bootstrapping algorithm in order to improve the probabilities P(v,c). We made a cursory examination of this technique but ran into difficulties due to the fact that the associations do not form valid probability distributions. We tried to fix this by normalizing association strengths (and also exp(A)) and distributing count mass among senses proportionally. However, even one additional iteration of the algorithm worsened our performance on the word sense disambiguation task. Further investigation into this strategy could involve getting probabilities of a sense given a verb in a more direct manner rather than using association strengths. Given such a strategy of dealing with disambiguation, another improvement would be to build a model that maps class to class relationships rather than word to class relationships. This would likely perform better, as one of the problems we noted was that some errors were caused by our model not distinguishing between the various senses of a verb. Finally, since the subject of the verb does have some influence on the direct object, as illustrated earlier, our model could be extended to take subject-verb-object triples rather than just verb-object pairs.

Bibliography:

Resnik P., Selectional Preference and Sense Disambiguation, in Proceedings of the ANLP-97 Workshop: Tagging Text with Lexical Semantics: Why, What, and How?, 1997.

Li and Abe, Clustering Word with the MDL Principle, in Proceedings of the 16th Conference on Computational Linguistics, 1998.

McCarthy D., Word Sense Disambiguation for Acquistion of Selectional Preferences, University of Sussex, 1997.

Manning and Schutze, Foundations of Statistical Natural Language Processing, The MIT Press, 1999.

Rooth et. al., Inducing a Semantically Annotated Lexicon via EM-based Clustering, ACL: Proceedings of the 37th Annual Meeting, 1999.