# AUTOMATIC ACQUISITION OF A LARGE SUBCATEGORIZATION DICTIONARY FROM CORPORA

**Christopher D. Manning**
*Xerox PARC and Stanford University*
Stanford University
Dept. of Linguistics, Bldg. 100
Stanford, CA 94305-2150, USA
Internet: manning@csli.stanford.edu

## Abstract

This paper presents a new method for producing a dictionary of subcategorization frames from unlabelled text corpora. It is shown that statistical filtering of the results of a finite state parser running on the output of a stochastic tagger produces high quality results, despite the error rates of the tagger and the parser. Further, it is argued that this method can be used to learn all subcategorization frames, whereas previous methods are not extensible to a general solution to the problem.

## INTRODUCTION

Rule-based parsers use subcategorization information to constrain the number of analyses that are generated. For example, from subcategorization alone, we can deduce that the PP in (1) must be an argument of the verb, not a noun phrase modifier:

(1) John put [$_{\text{NP}}$the cactus] [$_{\text{PP}}$on the table].

Knowledge of subcategorization also aids text generation programs and people learning a foreign language.

A subcategorization frame is a statement of what types of syntactic arguments a verb (or adjective) takes, such as objects, infinitives, *that*-clauses, participial clauses, and subcategorized prepositional phrases. In general, verbs and adjectives each appear in only a small subset of all possible argument subcategorization frames.

A major bottleneck in the production of high-coverage parsers is assembling lexical information, such as subcategorization information. In early and much continuing work in computational linguistics, this information has been coded laboriously by hand. More recently, on-line versions of dictionaries that provide subcategorization information have become available to researchers (Hornby 1989, Procter 1978, Sinclair 1987). But this is the same method of obtaining subcategorizations – painstaking work by hand. We have simply passed the need for tools that acquire lexical information from the computational linguist to the lexicographer.

Thus there is a need for a program that can acquire a subcategorization dictionary from on-line corpora of unrestricted text:

1. Dictionaries with subcategorization information are unavailable for most languages (only a few recent dictionaries, generally targeted at non-native speakers, list subcategorization frames).
2. No dictionary lists verbs from specialized subfields (as in *I telneted to Princeton*), but these could be obtained automatically from texts such as computer manuals.
3. Hand-coded lists are expensive to make, and invariably incomplete.
4. A subcategorization dictionary obtained automatically from corpora can be updated quickly and easily as different usages develop. Dictionaries produced by hand always substantially lag real language use.

The last two points do not argue against the use of existing dictionaries, but show that the incomplete information that they provide needs to be supplemented with further knowledge that is best collected automatically.[1] The desire to combine hand-coded and automatically learned knowledge

[0]Thanks to Julian Kupiec for providing the tagger on which this work depends and for helpful discussions and comments along the way. I am also indebted for comments on an earlier draft to Marti Hearst (whose comments were the most useful!), Hinrich Schütze, Penni Sibun, Mary Dalrymple, and others at Xerox PARC, where this research was completed during a summer internship; Stanley Peters, and the two anonymous ACL reviewers.

[1]A point made by Church and Hanks (1989). Arbitrary gaps in listing can be smoothed with a program such as the work presented here. For example, among the 27 verbs that most commonly cooccurred with *from*, Church and Hanks found 7 for which this subcategorization frame was not listed in the Cobuild

suggests that we should aim for a high precision learner (even at some cost in coverage), and that is the approach adopted here.

## DEFINITIONS AND DIFFICULTIES

Both in traditional grammar and modern syntactic theory, a distinction is made between arguments and adjuncts. In sentence (2), *John* is an argument and *in the bathroom* is an adjunct:

(2) Mary berated John in the bathroom.

Arguments fill semantic slots licensed by a particular verb, while adjuncts provide information about sentential slots (such as time or place) that can be filled for any verb (of the appropriate aspectual type).

While much work has been done on the argument/adjunct distinction (see the survey of distinctions in Pollard and Sag (1987, pp. 134–139)), and much other work presupposes this distinction, in practice, it gets murky (like many things in linguistics). I will adhere to a conventional notion of the distinction, but a tension arises in the work presented here when judgments of argument/adjunct status reflect something other than frequency of cooccurrence – since it is actually cooccurrence data that a simple learning program like mine uses. I will return to this issue later.

Different classifications of subcategorization frames can be found in each of the dictionaries mentioned above, and in other places in the linguistics literature. I will assume without discussion a fairly standard categorization of subcategorization frames into 19 classes (some parameterized for a preposition), a selection of which are shown below:

| | |
|---|---|
| IV | Intransitive verbs |
| TV | Transitive verbs |
| DTV | Ditransitive verbs |
| THAT | Takes a finite *that* complement |
| NPTHAT | Direct object and *that* complement |
| INF | Infinitive clause complement |
| NPINF | Direct object and infinitive clause |
| ING | Takes a participial VP complement |
| P(*prep*) | Prepositional phrase headed by *prep* |
| NP_P(*prep*) | Direct object and PP headed by *prep* |

dictionary (Sinclair 1987). The learner presented here finds a subcategorization involving *from* for all but one of these 7 verbs (the exception being *ferry* which was fairly rare in the training corpus).

## PREVIOUS WORK

While work has been done on various sorts of collocation information that can be obtained from text corpora, the only research that I am aware of that has dealt directly with the problem of the automatic acquisition of subcategorization frames is a series of papers by Brent (Brent and Berwick 1991, Brent 1991, Brent 1992). Brent and Berwick (1991) took the approach of trying to generate very high precision data.[2] The input was hand-tagged text from the Penn Treebank, and they used a very simple finite state parser which ignored nearly all the input, but tried to learn from the sentences which seemed least likely to contain false triggers – mainly sentences with pronouns and proper names.[3] This was a consistent strategy which produced promising initial results.

However, using hand-tagged text is clearly not a solution to the knowledge acquisition problem (as hand-tagging text is more laborious than collecting subcategorization frames), and so, in more recent papers, Brent has attempted learning subcategorizations from untagged text. Brent (1991) used a procedure for identifying verbs that was still very accurate, but which resulted in extremely low yields (it garnered as little as 3% of the information gained by his subcategorization learner running on tagged text, which itself ignored a huge percentage of the information potentially available). More recently, Brent (1992) substituted a very simple heuristic method to detect verbs (anything that occurs both with and without the suffix *-ing* in the text is taken as a potential verb, and every potential verb token is taken as an actual verb unless it is preceded by a determiner or a preposition other than *to*.[4] This is a rather simplistic and inadequate approach to verb detection, with a very high error rate. In this work I will use a stochastic part-of-speech tagger to detect verbs (and the part-of-speech of other words), and will suggest that this gives much better results.[5]

Leaving this aside, moving to either this last approach of Brent's or using a stochastic tagger undermines the consistency of the initial approach. Since the system now makes integral use of a high-error-rate component,[6] it makes little sense

[2] That is, data with very few errors.

[3] A false trigger is a clause in the corpus that one wrongly takes as evidence that a verb can appear with a certain subcategorization frame.

[4] Actually, learning occurs only from verbs in the base or *-ing* forms; others are ignored (Brent 1992, p. 8).

[5] See Brent (1992, p. 9) for arguments against using a stochastic tagger; they do not seem very persuasive (in brief, there is a chance of spurious correlations, and it is difficult to evaluate composite systems).

[6] On the order of a 5% error rate on each token for

for other components to be exceedingly selective about which data they use in an attempt to avoid as many errors as possible. Rather, it would seem more desirable to extract as much information as possible out of the text (even if it is noisy), and then to use appropriate statistical techniques to handle the noise.

There is a more fundamental reason to think that this is the right approach. Brent and Berwick's original program learned just five subcategorization frames (TV, THAT, NPTHAT, INF and NPINF). While at the time they suggested that "we foresee no impediment to detecting many more," this has apparently not proved to be the case (in Brent (1992) only six are learned: the above plus DTV). It seems that the reason for this is that their approach has depended upon finding cues that are very accurate predictors for a certain subcategorization (that is, there are very few false triggers), such as pronouns for NP objects and *to* plus a finite verb for infinitives. However, for many subcategorizations there just are no highly accurate cues.[7] For example, some verbs subcategorize for the preposition *in*, such as the ones shown in (3):

(3) a. Two women are *assisting* the police *in* their investigation.
 b. We *chipped in* to buy her a new TV.
 c. His letter was *couched in* conciliatory terms.

But the majority of occurrences of *in* after a verb are NP modifiers or non-subcategorized locative phrases, such as those in (4).[8]

(4) a. He gauged support for a change in the party leadership.
 b. He built a ranch in a new suburb.
 c. We were traveling along in a noisy helicopter.

There just is no high accuracy cue for verbs that subcategorize for *in*. Rather one must collect cooccurrence statistics, and use significance testing, a mutual information measure or some other form of statistic to try and judge whether a particular verb subcategorizes for *in* or just sometimes appears with a locative phrase.[9] Thus, the strategy I will use is to collect as much (fairly accurate) information as possible from the text corpus, and then use statistical filtering to weed out false cues.

## METHOD

One month (approximately 4 million words) of the New York Times newswire was tagged using a version of Julian Kupiec's stochastic part-of-speech tagger (Kupiec 1992).[10] Subcategorization learning was then performed by a program that processed the output of the tagger. The program had two parts: a finite state parser ran through the text, parsing auxiliary sequences and noting complements after verbs and collecting histogram-type statistics for the appearance of verbs in various contexts. A second process of statistical filtering then took the raw histograms and decided the best guess for what subcategorization frames each observed verb actually had.

### The finite state parser

The finite state parser essentially works as follows: it scans through text until it hits a verb or auxiliary, it parses any auxiliaries, noting whether the verb is active or passive, and then it parses complements following the verb until something recognized as a terminator of subcategorized arguments is reached.[11] Whatever has been found is entered in the histogram. The parser includes a simple NP recognizer (parsing determiners, possessives, adjectives, numbers and compound nouns) and various other rules to recognize certain cases that appeared frequently (such as direct quotations in either a normal or inverted, quotation first, order). The parser does not learn from participles since an NP after them may be the subject rather than the object (e.g., *the yawning man*).

The parser has 14 states and around 100 transitions. It outputs a list of elements occurring after the verb, and this list together with the record of whether the verb is passive yields the overall context in which the verb appears. The parser skips to the start of the next sentence in a few cases where things get complicated (such as on encountering a

the stochastic tagger (Kupiec 1992), and a presumably higher error rate on Brent's technique for detecting verbs.

[7] This inextensibility is also discussed by Hearst (1992).

[8] A sample of 100 uses of *in* from the New York Times suggests that about 70% of uses are in post-verbal contexts, but, of these, only about 15% are subcategorized complements (the rest being fairly evenly split between NP modifiers and time or place adjunct PPs).

[9] One cannot just collect verbs that always appear with *in* because many verbs have multiple subcategorization frames. As well as (3b), *chip* can also just be a TV: *John chipped his tooth.*

[10] Note that the input is very noisy text, including sports results, bestseller lists and all the other vagaries of a newswire.

[11] As well as a period, things like subordinating conjunctions mark the end of subcategorized arguments. Additionally, clausal complements such as those introduced by *that* function both as an argument and as a marker that this is the final argument.

conjunction, the scope of which is ambiguous, or a relative clause, since there will be a gap somewhere within it which would give a wrong observation). However, there are many other things that the parser does wrong or does not notice (such as reduced relatives). One could continue to refine the parser (up to the limits of what can be recognized by a finite state device), but the strategy has been to stick with something simple that works a reasonable percentage of the time and then to filter its results to determine what subcategorizations verbs actually have.

Note that the parser does not distinguish between arguments and adjuncts.[12] Thus the frame it reports will generally contain too many things. Indicative results of the parser can be observed in Fig. 1, where the first line under each line of text shows the frames that the parser found. Because of mistakes, skipping, and recording adjuncts, the finite state parser records nothing or the wrong thing in the majority of cases, but, nevertheless, enough good data are found that the final subcategorization dictionary describes the majority of the subcategorization frames in which the verbs are used in this sample.

### Filtering

Filtering assesses the frames that the parser found (called *cues* below). A cue may be a correct subcategorization for a verb, or it may contain spurious adjuncts, or it may simply be wrong due to a mistake of the tagger or the parser. The filtering process attempts to determine whether one can be highly confident that a cue which the parser noted is actually a subcategorization frame of the verb in question.

The method used for filtering is that suggested by Brent (1992). Let $B_s$ be an estimated upper bound on the probability that a token of a verb that doesn't take the subcategorization frame $s$ will nevertheless appear with a cue for $s$. If a verb appears $m$ times in the corpus, and $n$ of those times it cooccurs with a cue for $s$, then the probability that all the cues are false cues is bounded by the binomial distribution:

$$\sum_{i=n}^{m} \frac{m!}{i!(m-i)!} {B_s}^i (1-B_s)^{m-i}$$

Thus the null hypothesis that the verb does not have the subcategorization frame $s$ can be rejected if the above sum is less than some confidence level $C$ ($C = 0.02$ in the work reported here).

Brent was able to use extremely low values for $B_s$ (since his cues were sparse but unlikely to be false cues), and indeed found the best performance with values of the order of $2^{-8}$. However, using my parser, false cues are common. For example, when the recorded subcategorization is __ NP PP(of), it is likely that the PP should actually be attached to the NP rather than the verb. Hence I have used high bounds on the probability of cues being false cues for certain triggers (the used values range from 0.25 (for TV-P(*of*)) to 0.02). At the moment, the false cue rates $B_s$ in my system have been set empirically. Brent (1992) discusses a method of determining values for the false cue rates automatically, and this technique or some similar form of automatic optimization could profitably be incorporated into my system.

## RESULTS

The program acquired a dictionary of 4900 subcategorizations for 3104 verbs (an average of 1.6 per verb). Post-editing would reduce this slightly (a few repeated typos made it in, such as *acknowlege*, a few oddities such as the spelling *garontee* as a 'Cajun' pronunciation of *guarantee* and a few cases of mistakes by the tagger which, for example, led it to regard *lowlife* as a verb several times by mistake). Nevertheless, this size already compares favorably with the size of some production MT systems (for example, the English dictionary for Siemens' METAL system lists about 2500 verbs (Adriaens and de Braekeleer 1992)). In general, all the verbs for which subcategorization frames were determined are in Webster's (Gove 1977) (the only noticed exceptions being certain instances of prefixing, such as *overcook* and *repurchase*), but a larger number of the verbs do not appear in the only dictionaries that list subcategorization frames (as their coverage of words tends to be more limited). Examples are *fax*, *lambaste*, *skedaddle*, *sensationalize*, and *solemnize*. Some idea of the growth of the subcategorization dictionary can be had from Table 1.

Table 1. Growth of subcategorization dictionary

| Words Processed (million) | Verbs in subcat dictionary | Subcats learned | Subcats learned per verb |
|---|---|---|---|
| 1.2 | 1856 | 2661 | 1.43 |
| 2.9 | 2689 | 4129 | 1.53 |
| 4.1 | 3104 | 4900 | 1.58 |

The two basic measures of results are the information retrieval notions of recall and precision: How many of the subcategorization frames of the verbs were learned and what percentage of the things in the induced dictionary are correct? I have done some preliminary work to answer these questions.

[12]Except for the fact that it will only count the first of multiple PPs as an argument.

In the mezzanine, a man came with two sons and one baseball glove, like so many others there, in case,
[p(with)]
$^{\text{OK}}$IV

of course, a foul ball was hit to them. The father sat throughout the game with the
[pass,p(to)] [p(throughout)]
$^{\text{OK}}$TV *IV

glove on, leaning forward in anticipation like an outfielder before every pitch. By the sixth inning, he

*P(forward)

appeared exhausted from his exertion. The kids didn't seem to mind that the old man hogged the
[xcomp,p(from)] [inf] [that] [np]
*XCOMP $^{\text{OK}}$INF $^{\text{OK}}$THAT $^{\text{OK}}$TV

glove. They had their hands full with hot dogs. Behind them sat a man named Peter and his son
[that]
*TV-XCOMP *IV $^{\text{OK}}$DTV

Paul. They discussed the merits of Carreon over McReynolds in left field, and the advisability of
[np,p(of)]
$^{\text{OK}}$TV

replacing Cone with Musselman. At the seventh-inning stretch, Peter, who was born in Austria but

$^{\text{OK}}$TV-P(with) $^{\text{OK}}$TV

came to America at age 10, stood with the crowd as "Take Me Out to the Ball Game" was played. The

$^{\text{OK}}$P(to) $^{\text{OK}}$IV $^{\text{OK}}$TV

fans sang and waved their orange caps.
[np]
$^{\text{OK}}$IV $^{\text{OK}}$TV

Figure 1. A randomly selected sample of text from the New York Times, with what the parser could extract from the text on the second line and whether the resultant dictionary has the correct subcategorization for this occurrence shown on the third line ($^{\text{OK}}$ indicates that it does, while * indicates that it doesn't).

For recall, we might ask how many of the uses of verbs in a text are captured by our subcategorization dictionary. For two randomly selected pieces of text from other parts of the New York Times newswire, a portion of which is shown in Fig. 1, out of 200 verbs, the acquired subcategorization dictionary listed 163 of the subcategorization frames that appeared. So the token recall rate is approximately 82%. This compares with a baseline accuracy of 32% that would result from always guessing TV (transitive verb) and a performance figure of 62% that would result from a system that correctly classified all TV and THAT verbs (the two most common types), but which got everything else wrong.

We can get a pessimistic lower bound on precision and recall by testing the acquired dictionary against some published dictionary.[13] For this test, 40 verbs were selected (using a random number generator) from a list of 2000 common verbs.[14] Table 2 gives the subcategorizations listed in the OALD (recoded where necessary according to my classification of subcategorizations) and those in the subcategorization dictionary acquired by my program in a compressed format. Next to each verb, listing just a subcategorization frame means that it appears in both the OALD and my subcategorization dictionary, a subcategorization frame preceded by a minus sign (−) means that the subcategorization frame only appears in the OALD, and a subcategorization frame preceded by a plus sign (+) indicates one listed only in my program's subcategorization dictionary (i.e., one that is probably wrong).[15] The numbers are the number of cues that the program saw for each subcat-

[13] The resulting figures will be considerably lower than the true precision and recall because the dictionary lists subcategorization frames that do not appear in the training corpus and vice versa. However, this is still a useful exercise to undertake, as one can attain a high token success rate by just being able to accurately detect the most common subcategorization frames.

[14] The number 2000 is arbitrary, but was chosen following the intuition that one wanted to test the program's performance on verbs of at least moderate frequency.

[15] The verb *redesign* does not appear in the OALD, so its subcategorization entry was determined by me, based on the entry in the OALD for *design*.

egorization frame (that is in the resulting subcategorization dictionary). Table 3 then summarizes the results from the previous table. Lower bounds for the precision and recall of my induced subcategorization dictionary are approximately 90% and 43% respectively (looking at types).

The aim in choosing error bounds for the filtering procedure was to get a highly accurate dictionary at the expense of recall, and the lower bound precision figure of 90% suggests that this goal was achieved. The lower bound for recall appears less satisfactory. There is room for further work here, but this does represent a pessimistic lower bound (recall the 82% token recall figure above). Many of the more obscure subcategorizations for less common verbs never appeared in the modest-sized learning corpus, so the model had no chance to master them.[16]

Further, the learned corpus may reflect language use more accurately than the dictionary. The OALD lists *retire to NP* and *retire from NP* as subcategorized PP complements, but not *retire in NP*. However, in the training corpus, the collocation *retire in* is much more frequent than *retire to* (or *retire from*). In the absence of differential error bounds, the program is always going to take such more frequent collocations as subcategorized. Actually, in this case, this seems to be the right result. While *in* can also be used to introduce a locative or temporal adjunct:

(5) John retired from the army in 1945.

if *in* is being used similarly to *to* so that the two sentences in (6) are equivalent:

(6) a. John retired to Malibu.

b. John retired in Malibu.

it seems that *in* should be regarded as a subcategorized complement of *retire* (and so the dictionary is incomplete).

As a final example of the results, let us discuss verbs that subcategorize for *from* (cf. fn. 1 and Church and Hanks 1989). The acquired subcategorization dictionary lists a subcategorization involving *from* for 97 verbs. Of these, 1 is an outright mistake, and 1 is a verb that does not appear in the Cobuild dictionary (*reshape*). Of the rest, 64 are listed as occurring with *from* in Cobuild and 31 are not. While in some of these latter cases it could be argued that the occurrences of *from* are adjuncts rather than arguments, there are also

[16] For example, *agree about* did not appear in the learning corpus (and only once in total in another two months of the New York Times newswire that I examined). While *disagree about* is common, *agree about* seems largely disused: people like to agree *with* people but disagree *about* topics.

Table 2. Subcategorizations for 40 randomly selected verbs in OALD and acquired subcategorization dictionary (see text for key).

**agree:** INF:386, THAT:187, P(*to*):101, IV:77, P(*with*):79, P(*on*):63, −P(*about*), −WH

**ail:** −TV

**annoy:** −TV

**assign:** TV-P(*to*):19, NPINF:11, −TV-P(*for*), −DTV, +TV:7

**attribute:** TV-P(*to*):67, +P(*to*):12

**become:** TV:406, XCOMP:142, −PP(*of*)

**bridge:** TV:6, +P(*between*):3

**burden:** TV:6, TV-P(*with*):5

**calculate:** THAT:11, TV:4, −WH, −NPINF, −PP(*on*)

**chart:** TV:4, +DTV:4

**chop:** TV:4, −TV-P(*up*), −TV-P(*into*)

**depict:** TV-P(*as*):10, TV:9, −NPING

**dig:** TV:12, P(*out*):8, P(*up*):7, −IV, −TV-P(*in*), −TV-P(*out*), −TV-P(*over*), −TV-P(*up*), −P(*for*)

**drill:** TV-P(*in*):14, TV:14, −IV, −P(FOR)

**emanate:** P(*from*):2

**employ:** TV:31, −TV-P(*on*), −TV-P(*in*), −TV-P(*as*), −NPINF

**encourage:** NPINF:108, TV:60, −TV-P(*in*)

**exact:** −TV, −TV-PP(*from*)

**exclaim:** THAT:10, −IV, −P()

**exhaust:** TV:12

**exploit:** TV:11

**fascinate:** TV:17

**flavor:** TV:8, −TV-PP(*with*)

**heat:** IV:12, TV:9, −TV-P(*up*), −P(*up*)

**leak:** P(*out*):7, −IV, −P(*in*), −TV, −TV-P(*to*)

**lock:** TV:16, TV-P(*in*):16, −IV, −P(), −TV-P(*together*), −TV-P(*up*), −TV-P(*out*), −TV-P(*away*)

**mean:** THAT:280, TV:73, NPINF:57, INF:41, ING:35, −TV-PP(*to*), −POSSING, −TV-PP(*as*) −DTV, −TV-PP(*for*)

**occupy:** TV:17, −TV-P(*in*), −TV-P(*with*)

**prod:** TV:4, TV-P(*into*):3, −IV, −P(AT), −NPINF

**redesign:** TV:8, −TV-P(*for*), −TV-P(*as*), −NPINF

**reiterate:** THAT:13, −TV

**remark:** THAT:7, −P(*on*), −P(*upon*), −TV, +IV:3,

**retire:** IV:30, TV:9, −P(*from*), −P(*to*), −XCOMP, +P(*in*):38

**shed:** TV:8, −TV-P(*on*)

**sift:** P(*through*):8, −TV, −TV-P(OUT)

**strive:** INF:14, P(*for*):9, −P(*after*), −P(*against*), −P(*with*), −IV

**tour:** TV:9, IV:6, −P(IN)

**troop:** −IV, −P(), [TV: trooping the color]

**wallow:** P(*in*):2, −IV, −P(*about*), −P(*around*)

**water:** TV:13, −IV, −TV-P(*down*), +THAT:6

Table 3. Comparison of results with OALD

| | Subcategorization frames | | | |
|---|---|---|---|---|
| Word | Right | Wrong | Out of | Incorrect |
| agree: | 6 | | 8 | |
| ail: | 0 | | 1 | |
| annoy: | 0 | | 1 | |
| assign: | 2 | 1 | 4 | TV |
| attribute: | 1 | 1 | 1 | P(*to*) |
| become: | 2 | | 3 | |
| bridge: | 1 | 1 | 1 | TV-P(*between*) |
| burden: | 2 | | 2 | |
| calculate: | 2 | | 5 | |
| chart: | 1 | 1 | 1 | DTV |
| chop: | 1 | | 3 | |
| depict: | 2 | | 3 | |
| dig: | 3 | | 9 | |
| drill: | 2 | | 4 | |
| emanate: | 1 | | 1 | |
| employ: | 1 | | 5 | |
| encourage: | 2 | | 3 | |
| exact: | 0 | | 2 | |
| exclaim: | 1 | | 3 | |
| exhaust: | 1 | | 1 | |
| exploit: | 1 | | 1 | |
| fascinate: | 1 | | 1 | |
| flavor: | 1 | | 2 | |
| heat: | 2 | | 4 | |
| leak: | 1 | | 5 | |
| lock: | 2 | | 8 | |
| mean: | 5 | | 10 | |
| occupy: | 1 | | 3 | |
| prod: | 2 | | 5 | |
| redesign: | 1 | | 4 | |
| reiterate: | 1 | | 2 | |
| remark: | 1 | 1 | 4 | IV |
| retire: | 2 | 1 | 5 | P(*in*) |
| shed: | 1 | | 2 | |
| sift: | 1 | | 3 | |
| strive: | 2 | | 6 | |
| tour: | 2 | | 3 | |
| troop: | 0 | | 3 | |
| wallow: | 1 | | 4 | |
| water: | 1 | 1 | 3 | THAT |
| | 60 | 7 | 139 | |

Precision (percent right of ones learned): 90%
Recall (percent of OALD ones learned): 43%

some unquestionable omissions from the dictionary. For example, Cobuild does not list that *forbid* takes *from*-marked participial complements, but this is very well attested in the New York Times newswire, as the examples in (7) show:

(7) a. The Constitution appears to *forbid* the general, as a former president who came to power through a coup, *from* taking office.

b. Parents and teachers are *forbidden from* taking a lead in the project, and ...

Unfortunately, for several reasons the results presented here are not directly comparable with those of Brent's systems.[17] However, they seems to represent at least a comparable level of performance.

## FUTURE DIRECTIONS

This paper presented one method of learning subcategorizations, but there are other approaches one might try. For disambiguating whether a PP is subcategorized by a verb in the V NP PP environment, Hindle and Rooth (1991) used a *t*-score to determine whether the PP has a stronger association with the verb or the preceding NP. This method could be usefully incorporated into my parser, but it remains a special-purpose technique for one particular case. Another research direction would be making the parser stochastic as well, rather than it being a categorical finite state device that runs on the output of a stochastic tagger.

There are also some linguistic issues that remain. The most troublesome case for any English subcategorization learner is dealing with prepositional complements. As well as the issues discussed above, another question is how to represent the subcategorization frames of verbs that take a range of prepositional complements (but not all). For example, *put* can take virtually any locative or directional PP complement, while *lean* is more choosy (due to facts about the world):

[17] My system tries to learn many more subcategorization frames, most of which are more difficult to detect accurately than the ones considered in Brent's work, so overall figures are not comparable. The recall figures presented in Brent (1992) gave the rate of recall out of those verbs which generated at least one cue of a given subcategorization rather than out of all verbs that have that subcategorization (pp. 17–19), and are thus higher than the true recall rates from the corpus (observe in Table 3 that no cues were generated for infrequent verbs or subcategorization patterns). In Brent's earlier work (Brent 1991), the error rates reported were for learning from tagged text. No error rates for running the system on untagged text were given and no recall figures were given for either system.

(8) a. John leaned against the wall
b. *John leaned under the table
c. *John leaned up the chute

The program doesn't yet have a good way of representing classes of prepositions.

The applications of this system are fairly obvious. For a parsing system, the current subcategorization dictionary could probably be incorporated as is, since the utility of the increase in coverage would almost undoubtedly outweigh problems arising from the incorrect subcategorization frames in the dictionary. A lexicographer would want to review the results by hand. Nevertheless, the program clearly finds gaps in printed dictionaries (even ones prepared from machine-readable corpora, like Cobuild), as the above example with *forbid* showed. A lexicographer using this program might prefer it adjusted for higher recall, even at the expense of lower precision. When a seemingly incorrect subcategorization frame is listed, the lexicographer could then ask for the cues that led to the postulation of this frame, and proceed to verify or dismiss the examples presented.

A final question is the applicability of the methods presented here to other languages. Assuming the existence of a part-of-speech lexicon for another language, Kupiec's tagger can be trivially modified to tag other languages (Kupiec 1992). The finite state parser described here depends heavily on the fairly fixed word order of English, and so precisely the same technique could only be employed with other fixed word order languages. However, while it is quite unclear how Brent's methods could be applied to a free word order language, with the method presented here, there is a clear path forward. Languages that have free word order employ either case markers or agreement affixes on the head to mark arguments. Since the tagger provides this kind of morphological knowledge, it would be straightforward to write a similar program that determines the arguments of a verb using any combination of word order, case marking and head agreement markers, as appropriate for the language at hand. Indeed, since case-marking is in some ways more reliable than word order, the results for other languages might even be better than those reported here.

## CONCLUSION

After establishing that it is desirable to be able to automatically induce the subcategorization frames of verbs, this paper examined a new technique for doing this. The paper showed that the technique of trying to learn from easily analyzable pieces of data is not extendable to all subcategorization frames, and, at any rate, the sparseness of appropriate cues in unrestricted texts suggests that a better strategy is to try and extract as much (noisy) information as possible from as much of the data as possible, and then to use statistical techniques to filter the results. Initial experiments suggest that this technique works at least as well as previously tried techniques, and yields a method that can learn all the possible subcategorization frames of verbs.

## REFERENCES

Adriaens, Geert, and Gert de Braekeleer. 1992. Converting Large On-line Valency Dictionaries for NLP Applications: From PROTON Descriptions to METAL Frames. In *Proceedings of COLING-92*, 1182–1186.

Brent, Michael R. 1991. Automatic Acquisition of Subcategorization Frames from Untagged Text. In *Proceedings of the 29th Annual Meeting of the ACL*, 209–214.

Brent, Michael R. 1992. Robust Acquisition of Subcategorizations from Unrestricted Text: Unsupervised Learning with Syntactic Knowledge. MS, John Hopkins University, Baltimore, MD.

Brent, Michael R., and Robert Berwick. 1991. Automatic Acquisition of Subcategorization Frames from Free Text Corpora. In *Proceedings of the 4th DARPA Speech and Natural Language Workshop*. Arlington, VA: DARPA.

Church, Kenneth, and Patrick Hanks. 1989. Word Association Norms, Mutual Information, and Lexicography. In *Proceedings of the 27th Annual Meeting of the ACL*, 76–83.

Gove, Philip B. (ed.). 1977. *Webster's seventh new collegiate dictionary*. Springfield, MA: G. & C. Merriam.

Hearst, Marti. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of COLING-92*, 539–545.

Hindle, Donald, and Mats Rooth. 1991. Structural Ambiguity and Lexical Relations. In *Proceedings of the 29th Annual Meeting of the ACL*, 229–236.

Hornby, A. S. 1989. *Oxford Advanced Learner's Dictionary of Current English*. Oxford: Oxford University Press. 4th edition.

Kupiec, Julian M. 1992. Robust Part-of-Speech Tagging Using a Hidden Markov Model. *Computer Speech and Language* 6:225–242.

Pollard, Carl, and Ivan A. Sag. 1987. *Information-Based Syntax and Semantics*. Stanford, CA: CSLI.

Procter, Paul (ed.). 1978. *Longman Dictionary of Contemporary English*. Burnt Mill, Harlow, Essex: Longman.

Sinclair, John M. (ed.). 1987. *Collins Cobuild English Language Dictionary*. London: Collins.