AUTOMATIC ACQUISITION OF SUBCATEGORIZATION FRAMES FROM UNTAGGED TEXT

Michael R. Brent
MIT AI Lab
545 Technology Square
Cambridge, Massachusetts 02139
michael@ai.mit.edu

ABSTRACT

This paper describes an implemented program that takes a raw, untagged text corpus as its only input (no open-class dictionary) and generates a partial list of verbs occurring in the text and the subcategorization frames (SFs) in which they occur. Verbs are detected by a novel technique based on the Case Filter of Rouvret and Vergnaud (1980). The completeness of the output list increases monotonically with the total number of occurrences of each verb in the corpus. False positive rates are one to three percent of observations. Five SFs are currently detected and more are planned. Ultimately, I expect to provide a large SF dictionary to the NLP community and to train dictionaries for specific corpora.

1 INTRODUCTION

This paper describes an implemented program that takes an untagged text corpus and generates a partial list of verbs occurring in it and the subcategorization frames (SFs) in which they occur. So far, it detects the five SFs shown in Table 1.

SF Description	Good Example	Bad Example
direct object	greet them	*arrive them
direct object & clause	tell him he's a fool	*hope him he's a fool
direct object & infinitive	want him to attend	*hope him to attend
clause	know I'll attend	*want I'll attend
infinitive	hope to attend	*greet to attend

Table 1: The five subcategorization frames (SFs) detected so far

The SF acquisition program has been tested on a corpus of 2.6 million words of the Wall Street Journal (kindly provided by the Penn Tree Bank project). On this corpus, it makes 5101 observations about 2258 orthographically distinct verbs. False positive rates vary from one to three percent of observations, depending on the SF.

1.1 WHY IT MATTERS

Accurate parsing requires knowing the subcategorization frames of verbs, as shown by (1).

(1) a. I expected [$_{NP}$ the man who smoked $_{NP}$] to eat ice-cream
b. I doubted [$_{NP}$ the man who liked to eat ice-cream $_{NP}$]

Current high-coverage parsers tend to use either custom, hand-generated lists of subcategorization frames (e.g., Hindle, 1983), or published, hand-generated lists like the *Oxford Advanced Learner's Dictionary of Contemporary English*, Hornby and Covey (1973) (e.g., DeMarcken, 1990). In either case, such lists are expensive to build and to maintain in the face of evolving usage. In addition, they tend not to include rare usages or specialized vocabularies like financial or military jargon. Further, they are often incomplete in arbitrary ways. For example, Webster's Ninth New Collegiate Dictionary lists the sense of *strike* meaning "to occur to", as in "it struck him that...", but it does not list that same sense of *hit*. (My program discovered both.)

1.2 WHY IT'S HARD

The initial priorities in this research were:

- Generality (e.g., minimal assumptions about the text)
- Accuracy in identifying SF occurrences
- Simplicity of design and speed

Efficient use of the available text was not a high priority, since it was felt that plenty of text was available even for an inefficient learner, assuming sufficient speed to make use of it. These priorities

had a substantial influence on the approach taken. They are evaluated in retrospect in Section 4.

The first step in finding a subcategorization frame is finding a verb. Because of widespread and productive noun/verb ambiguity, dictionaries are not much use — they do not reliably exclude the possibility of lexical ambiguity. Even if they did, a program that could only learn SFs for unambiguous verbs would be of limited value. Statistical disambiguators make dictionaries more useful, but they have a fairly high error rate, and degrade in the presence of many unfamiliar words. Further, it is often difficult to understand where the error is coming from or how to correct it. So finding verbs poses a serious challenge for the design of an accurate, general-purpose algorithm for detecting SFs.

In fact, finding main verbs is more difficult than it might seem. One problem is distinguishing participles from adjectives and nouns, as shown below.

(2) a. John has [$_{NP}$ rented furniture]
(comp.: John has often rented apartments)
b. John was smashed (drunk) last night
(comp.: John was kissed last night)
c. John's favorite activity is watching TV
(comp.: John's favorite child is watching TV)

In each case the main verb is *have* or *be* in a context where most parsers (and statistical disambiguators) would mistake it for an auxiliary and mistake the following word for a participial main verb.

A second challenge to accuracy is determining which verb to associate a given complement with. Paradoxically, example (1) shows that in general it isn't possible to do this without already knowing the SF. One obvious strategy would be to wait for sentences where there is only one candidate verb; unfortunately, it is very difficult to know for certain how many verbs occur in a sentence. Finding some of the verbs in a text reliably is hard enough; finding all of them reliably is well beyond the scope of this work.

Finally, any system applied to real input, no matter how carefully designed, will occasionally make errors in finding the verb and determining its subcategorization frame. The more times a given verb appears in the corpus, the more likely it is that one of those occurrences will cause an erroneous judgment. For that reason any learning system that gets only positive examples and makes a permanent judgment on a single example will always degrade as the number of occurrences increases. In fact, making a judgment based on any fixed number of examples with any finite error rate will always lead to degradation with corpus-size. A better approach is to require a fixed percentage of the total occurrences of any given verb to appear with a given SF before concluding that random error is not responsible for these observations. Unfortunately, determining the cutoff percentage requires human intervention and sampling error makes classification unstable for verbs with few occurrences in the input. The sampling error can be dealt with (Brent, 1991) but predetermined cutoff percentages still require eye-balling the data. Thus robust, unsupervised judgments in the face of error pose the third challenge to developing an accurate learning system.

1.3 HOW IT'S DONE

The architecture of the system, and that of this paper, directly reflects the three challenges described above. The system consists of three modules:

1. Verb detection: Finds some occurrences of verbs using the Case Filter (Rouvret and Vergnaud, 1980), a proposed rule of grammar.
2. SF detection: Finds some occurrences of five subcategorization frames using a simple, finite-state grammar for a fragment of English.
3. SF decision: Determines whether a verb is genuinely associated with a given SF, or whether instead its apparent occurrences in that SF are due to error. This is done using statistical models of the frequency distributions.

The following two sections describe and evaluate the verb detection module and the SF detection module, respectively; the decision module, which is still being refined, will be described in a subsequent paper. The final two sections provide a brief comparison to related work and draw conclusions.

2 VERB DETECTION

The technique I developed for finding verbs is based on the Case Filter of Rouvret and Vergnaud (1980). The Case Filter is a proposed rule of grammar which, as it applies to English, says that every noun-phrase must appear either immediately to the left of a tensed verb, immediately to the right of a preposition, or immediately to the right of a main verb. Adverbs and adverbial phrases (including days and dates) are ignored for the purposes of case adjacency. A noun-phrase that satisfies the Case Filter is said to "get case" or "have case", while one that violates it is said to "lack case". The program judges an open-class word to be a main verb if it is adjacent to a pronoun or proper name that would otherwise lack case. Such a pronoun or proper name is either the subject or

the direct object of the verb. Other noun phrases are not used because it is too difficult to determine their right boundaries accurately.

The two criteria for evaluating the performance of the main-verb detection technique are efficiency and accuracy. Both were measured using a 2.6 million word corpus for which the Penn Treebank project provides hand-verified tags.

Efficiency of verb detection was assessed by running the SF detection module in the normal mode, where verbs were detected using the Case Filter technique, and then running it again with the Penn Tags substituted for the verb detection module. The results are shown in Table 2. Note

SF	Occurrences Found	Control	Efficiency
direct object	3,591	8,606	40%
direct object & clause	94	381	25%
direct object & infinitive	310	3,597	8%
clause	739	14,144	5%
infinitive	367	11,880	3%

Table 2: Efficiency of verb detection for each of the five SFs, as tested on 2.6 million words of the Wall Street Journal and controlled by the Penn Treebank's hand-verified tagging

the substantial variation among the SFs: for the SFs "direct object" and "direct object & clause" efficiency is roughly 40% and 25%, respectively; for "direct object & infinitive" it drops to about 8%; and for the intransitive SFs it is under 5%. The reason that the transitive SFs fare better is that the direct object gets case from the preceding verb and hence reveals its presence — intransitive verbs are harder to find. Likewise, clauses fare better than infinitives because their subjects get case from the main verb and hence reveal it, whereas infinitives lack overt subjects. Another obvious factor is that, for every SF listed above except "direct object" two verbs need to be found — the matrix verb and the complement verb — if either one is not detected then no observation is recorded.

Accuracy was measured by looking at the Penn tag for every word that the system judged to be a verb. Of approximately 5000 verb tokens found by the Case Filter technique, there were 28 disagreements with the hand-verified tags. My program was right in 8 of these cases and wrong in 20, for a 0.24% error-rate beyond the rate using hand-verified tags. Typical disagreements in which my system was right involved verbs that are ambiguous with much more frequent nouns, like *mold* in "The Soviet Communist Party has the power to shape corporate development and mold it into a body dependent upon it ." There were several systematic constructions in which the Penn tags were right and my system was wrong, including constructions like "We consumers are..." and pseudo-clefts like "what you then do is you make them think.... (These examples are actual text from the Penn corpus.)

The extraordinary accuracy of verb detection — within a tiny fraction of the rate achieved by trained human taggers — and it's relatively low efficiency are consistent with the priorities laid out in Section 1.2.

2.1 SF DETECTION

The obvious approach to finding SFs like "V NP to V" and "V to V" is to look for occurrences of just those patterns in the training corpus; but the obvious approach fails to address the attachment problem illustrated by example (1) above. The solution is based on the following insights:

- Some examples are clear and unambiguous.
- Observations made in clear cases generalize to all cases.
- It is possible to distinguish the clear cases from the ambiguous ones with reasonable accuracy.
- With enough examples, it pays to wait for the clear cases.

Rather than take the obvious approach of looking for "V NP to V", my approach is to wait for clear cases like "V PRONOUN to V". The advantages can be seen by contrasting (3) with (1).

(3) a. OK I expected him to eat ice-cream
b. * I doubted him to eat ice-cream

More generally, the system recognizes linguistic structure using a small finite-state grammar that describes only that fragment of English that is most useful for recognizing SFs. The grammar relies exclusively on closed-class lexical items such as pronouns, prepositions, determiners, and auxiliary verbs.

The grammar for detecting SFs needs to distinguish three types of complements: direct objects, infinitives, and clauses. The grammars for each of these are presented in Figure 1. Any open-class word judged to be a verb (see Section 2) and followed immediately by matches for <DO>, <clause>, <infinitive>, <DO><clause>, or <DO><inf> is assigned the corresponding SF. Any word ending in "ly" or

```
<clause>        := that? (<subj-pron> | <subj-obj-pron> | his | <proper-name>)
                   <tensed-verb>
<subj-pron>     := I | he | she | I | they
<subj-obj-pron> := you, it, yours, hers, ours, theirs
<DO>            := <obj-pron>
<obj-pron>      := me | him | us | them
<infinitive>    := to <previously-noted-uninflected-verb>
```

Figure 1: A non-recursive (finite-state) grammar for detecting certain verbal complements. "?" indicates an optional element. Any verb followed immediately expressions matching <DO>, <clause>, <infinitive>, <DO> <clause>, or <DO> <infinitive> is assigned the corresponding SF.

belonging to a list of 25 irregular adverbs is ignored for purposes of adjacency. The notation "?" follows optional expressions. The category **previously-noted-uninflected-verb** is special in that it is not fixed in advance — open-class non-adverbs are added to it when they occur following an unambiguous modal.[1] This is the only case in which the program makes use of earlier decisions — literally bootstrapping. Note, however, that ambiguity is possible between mass nouns and uninflected verbs, as in *to fish*.

Like the verb detection algorithm, the SF detection algorithm is evaluated in terms of efficiency and accuracy. The most useful estimate of efficiency is simply the density of observations in the corpus, shown in the first column of Table 3. The accuracy of SF detection is shown in the second column of Table 3.[2] The most common source of error was purpose adjuncts, as in "John quit to pursue a career in finance," which comes from omitting the *in order* from "John quit *in order* to pursue a career in finance." These purpose adjuncts were mistaken for infinitival complements. The other errors were more sporadic in nature, many coming from unusual extrapositions or other relatively rare phenomena.

SF	occurrences found	% error
direct object	3,591	1.5%
direct object & clause	94	2.0%
direct object & infinitive	310	1.5%
clause	739	0.5%
infinitive	367	3.0%

Table 3: SF detector error rates as tested on 2.6 million words of the Wall Street Journal

Once again, the high accuracy and low efficiency are consistent with the priorities of Section 1.2. The throughput rate is currently about ten-thousand words per second on a Sparcstation 2, which is also consistent with the initial priorities. Furthermore, at ten-thousand words per second the current density of observations is not problematic.

3 RELATED WORK

Interest in extracting lexical and especially collocational information from text has risen dramatically in the last two years, as sufficiently large corpora and sufficiently cheap computation have become available. Three recent papers in this area are Church and Hanks (1990), Hindle (1990), and Smadja and McKeown (1990). The latter two are concerned exclusively with collocation relations between open-class words and not with grammatical properties. Church is also interested primarily in open-class collocations, but he does discuss verbs that tend to be followed by infinitives within his mutual information framework.

Mutual information, as applied by Church, is a measure of the tendency of two items to appear near one-another — their observed frequency in nearby positions is divided by the expectation of that frequency if their positions were random and independent. To measure the tendency of a verb to be followed within a few words by an infinitive, Church uses his statistical disambiguator

[1] If there were room to store an unlimited number of uninflected verbs for later reference then the grammar formalism would not be finite-state. In fact, a fixed amount of storage, sufficient to store all the verbs in the language, is allocated. This question is purely academic, however — a hash-table gives constant-time average performance.

[2] Error rates computed by hand verification of 200 examples for each SF using the tagged mode. These are estimated independently of the error rates for verb detection.

(Church, 1988) to distinguish between *to* as an infinitive marker and *to* as a preposition. Then he measures the mutual information between occurrences of the verb and occurrences of infinitives following within a certain number of words. Unlike our system, Church's approach does not aim to decide whether or not a verb occurs with an infinitival complement — example (1) showed that being followed by an infinitive is not the same as taking an infinitival complement. It might be interesting to try building a verb categorization scheme based on Church's mutual information measure, but to the best of our knowledge no such work has been reported.

4 CONCLUSIONS

The ultimate goal of this work is to provide the NLP community with a substantially complete, automatically updated dictionary of subcategorization frames. The methods described above solve several important problems that had stood in the way of that goal. Moreover, the results obtained with those methods are quite encouraging. Nonetheless, two obvious barriers still stand on the path to a fully automated SF dictionary: a decision algorithm that can handle random error, and techniques for detecting many more types of SFs.

Algorithms are currently being developed to resolve raw SF observations into genuine lexical properties and random error. The idea is to automatically generate statistical models of the sources of error. For example, purpose adjuncts like "John quit to pursue a career in finance" are quite rare, accounting for only two percent of the apparent infinitival complements. Furthermore, they are distributed across a much larger set of matrix verbs than the true infinitival complements, so any given verb should occur with a purpose adjunct extremely rarely. In a histogram sorting verbs by their apparent frequency of occurrence with infinitival complements, those that in fact have appeared with purpose adjuncts and not true subcategorized infinitives will be clustered at the low frequencies. The distributions of such clusters can be modeled automatically and the models used for identifying false positives.

The second requirement for automatically generating a full-scale dictionary is the ability to detect many more types of SFs. SFs involving certain prepositional phrases are particularly challenging. For example, while purpose adjuncts (mistaken for infinitival complements) are relatively rare, instrumental adjuncts as in "John hit the nail *with a hammer*" are more common. The problem, of course, is how to distinguish them from genuine, subcategorized PPs headed by *with*, as in "John sprayed the lawn with distilled water". The hope is that a frequency analysis like the one planned for purpose adjuncts will work here as well, but how successful it will be, and if successful how large a sample size it will require, remain to be seen.

The question of sample size leads back to an evaluation of the initial priorities, which favored simplicity, speed, and accuracy, over efficient use of the corpus. There are various ways in which the high-priority criteria can be traded off against efficiency. For example, consider (2c): one might expect that the overwhelming majority of occurrences of "is V-ing" are genuine progressives, while a tiny minority are cases copula. One might also expect that the occasional copula constructions are not concentrated around any one present participle but rather distributed randomly among a large population. If those expectations are true then a frequency-modeling mechanism like the one being developed for adjuncts ought to prevent the mistaken copula from doing any harm. In that case it might be worthwhile to admit "is V-ing", where V is known to be a (possibly ambiguous) verb root, as a verb, independent of the Case Filter mechanism.

ACKNOWLEDGMENTS

Thanks to Don Hindle, Lila Gleitman, and Jane Grimshaw for useful and encouraging conversations. Thanks also to Mark Liberman, Mitch Marcus and the Penn Treebank project at the University of Pennsylvania for supplying tagged text. This work was supported in part by National Science Foundation grant DCR-85552543 under a Presidential Young Investigator Award to Professor Robert C. Berwick.

References

[Brent, 1991] M. Brent. Semantic Classification of Verbs from their Syntactic Contexts: An Implemented Classifier for Stativity. In *Proceedings of the 5th European ACL Conference*. Association for Computational Linguistics, 1991.

[Church and Hanks, 1990] K. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Comp. Ling.*, 16, 1990.

[Church, 1988] K. Church. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of the 2nd ACL Conference on Applied NLP*. ACL, 1988.

[DeMarcken, 1990] C. DeMarcken. Parsing the LOB Corpus. In *Proceedings of the ACL*. Assocation for Comp. Ling., 1990.

[Gleitman, 1990] L. Gleitman. The structural sources of verb meanings. *Language Acquisition*, 1(1):3–56, 1990.

[Hindle, 1983] D. Hindle. User Manual for Fidditch, a Deterministic Parser. Technical Report 7590-142, Naval Research Laboratory, 1983.

[Hindle, 1990] D. Hindle. Noun classification from predicate argument structures. In *Proceedings of the 28th Annual Meeting of the ACL*, pages 268–275. ACL, 1990.

[Hornby and Covey, 1973] A. Hornby and A. Covey. *Oxford Advanced Learner's Dictionary of Contemporary English*. Oxford University Press, Oxford, 1973.

[Levin, 1989] B. Levin. English Verbal Diathesis. Lexicon Project orking Papers no. 32, MIT Center for Cognitive Science, MIT, Cambridge, MA., 1989.

[Pinker, 1989] S. Pinker. *Learnability and Cognition: The Acquisition of Argument Structure*. MIT Press, Cambridge, MA, 1989.

[Rouvret and Vergnaud, 1980] A. Rouvret and J-R Vergnaud. Specifying Reference to the Subject. *Linguistic Inquiry*, 11(1), 1980.

[Smadja and McKeown, 1990] F. Smadja and K. McKeown. Automatically extracting and representing collocations for language generation. In *28th Annual Meeting of the Association for Comp. Ling.*, pages 252–259. ACL, 1990.

[Zwicky, 1970] A. Zwicky. In a Manner of Speaking. *Linguistic Inquiry*, 2:223–233, 1970.