# Categorial Disambiguation

**Gavan Duffy**
Department of Government
The University of Texas at Austin
Austin, Texas 78712
ARPAnet: AI.Duffy@R20.UTEXAS.EDU

**Abstract**

This paper presents an implemented, computationally inexpensive technique for disambiguating categories (parts of speech) by exploiting constraints on possible category combinations. Early resolutions of category ambiguities provide a great deal of leverage, simplifying later resolutions of other types of lexical ambiguity.

## 1 Introduction

Ambiguities pervade natural language. Many strategies exist for resolving many varieties of ambiguity, including phrasal and clausal attachment ambiguities, anaphor ambiguities, referential ambiguities, and sense ambiguities. Categorial ambiguities arise when the lexical entry for a token (word or morpheme) indicates that the token may be given alternative category (part of speech) assignments, depending upon the context of usage.

Few strategies exist for resolving categorial ambiguities, the simplest lexical ambiguity of all. Existing approaches resolve categorial ambiguities either (a) by following all categorial parsing paths until a grammatical path terminates, or (b) as part of the process of resolving sense ambiguities. Both approaches are computationally expensive. This paper presents an alternative approach that has been implemented in a working English-language parser [1].

The categorial disambiguator described here constitutes a simple method for resolving categorial ambiguities before phrase structures are created. As a result, only one parsing path need be followed and any later sense disambiguation is greatly simplified.

## 2 An Overview of Categorial Disambiguation

The categorial disambiguator assigns the appropriate category (part of speech) to a word whenever more than one such assignment is possible. For example, in sentence (1) below, the words "doctor", "might", "cure", and "patient" are each categorially ambiguous. Both "doctor" and "cure" could be a noun or a verb. "Might" could be a noun or a verb auxiliary. And "patient" could be an adjective or a noun. In (1), only the two instances of the definite article "the" are not categorially ambiguous. Yet, as we shall see, these suffice to disambiguate the category assignments for the entire sentence.

**(1) The doctor might cure the patient.**

When considering a design for the categorial disambiguator, an immediate inspiration was Waltz' [4] constraint-propagation approach for detecting legal junctions in line-drawings. Applying this approach to the detection of legal category combinations in English sentences proved to be quite straightforward.

Prioritized pattern action rules (CONDs) are specified for each known categorial ambiguity. Approximately 40 such rules are currently in place. Each rule is passed lists of the words and categories preceding and succeeding a categorial ambiguity. Other categorial ambiguities are represented as lists embedded within these preceding and succeeding category lists. Unambiguous categories are represented as symbols in those lists. Additionally, each rule knows the current word and maintains a list of its possible category assignments, which had previously been extracted from a lexicon.

When a disambiguation rule succeeds, it propagates its resolution as a constraint for disambiguating neighboring ambiguities. When a rule fails to resolve the ambiguity, the ambiguous alternatives remain in the category lists. When subsequent disambiguations propagate additional constraint, the disambiguation rule for this ambiguity is re-evaluated.

Ordinarily, disambiguation rules need examine only the categories of its sentential neighbors. These usually provide sufficient constraint to select one correct interpretation. Occasionally, however, the words in a sentence must be consulted in addition to their possible category assignments. For example, deciding whether a particular word is an adjective sometimes depends upon knowing whether a preceding verb accepts predicate adjective arguments. For maximal flexibility, disambiguation rules are free to query the lexicon about syntactic and semantic properties (subcategories) of particular words.

## 3 An Example of Categorial Disambiguation

The disambiguator resolves sentence (1) in the following steps:

1. "The" is unambiguously a determiner.
2. "Doctor" cannot be a verb because it follows the determiner "the". It must therefore be a noun.
3. "Might" might attach to "doctor" genitively, just as "door" sometimes attaches to "car" to form "car door". Since we are working without information regarding sense, we cannot reject this possibility. Thus, "might" remains categorially ambiguous. It is either a noun or a verb auxiliary.

4. "Cure" as a verb is consistent with the interpretation of "might" as a verb auxiliary. If "might" were a noun, then "cure" could not be a verb, since it disagrees in number with "might" as its subject. Of course "doctor might cure" could itself be a genitive formation, much the same as "car door handle". But if this were the case, then the next word, "the", would be anomalous. Two independent noun phrases rarely appear before a verb. "Cure" must therefore be a verb. Since "cure" is a verb, "might" must be a verb-auxiliary. Otherwise, the number of "cure" would be inconsistent with the number of either "doctor" or "might" as the clausal subject.

5. "The", again, is unambiguous.

6. "Patient" follows the determiner "the", so it could still be either a noun or an adjective. However, "patient" terminates the current clause, so the analysis of "patient" as a noun is preferred.

## 4 The Fallibility of Categorial Disambiguation

The careful reader will have noticed that categorial disambiguation is fallible. Two possible sources of error appear in sentence (1). First, "cure" might indeed be a noun, since two independent noun phrases sometimes *do* appear before a verb phrase, when a complement has been deleted, as in sentence (2).

**(2) Men dogs bite scream.**

In (1), the rule that disambiguates "cure" can check the current clause for other possible verbs. Since "doctor" has already been disambiguated as a noun, no other possible verbs remain in the clause. Thus, the interpretation of "cure" as a verb is preferred.

In (2), when the rule which disambiguates "dogs" (which might be a noun or a verb) examines the clausal environment, it notices that two verbs – "bite" and "scream" are available in the current clause, warranting the nominal interpretation of "dogs".

Actually, this case is somewhat more complex, since both "bite" and "scream" are themselves categorially ambiguous. Each could be either a noun or a verb. The ambiguity is resolvable, however. Since "men" is unambiguously a noun in (2), and since the three remaining tokens are all doubly ambiguous, there are $2^3 = 8$ possible sequences of noun phrases (NP) and verb phrases (VP) in this sentence. Only one of these sequences – NP NP VP VP with a deleted complement between the two NPs – is grammatical.

Both of these examples involve secondary searches through the clause. These occur only for a very limited range of cases. For example, had the verb auxiliary in (1) been "would", instead of "might", such a search would not be needed. In (2) the presence of an auxiliary, a determiner, or a cpomplement might obviate such a secondary search. On the basis of a worst-case analysis, one might consider this practice explosive, but since such secondary searches are rare and since most clauses are not extremely long, the disambiguation procedures terminate quickly in practice.

The second possible source of error involves sense ambiguities masquerading as categorial ambiguities. For example, the interpretation of "patient" as an adjective in (1) might certainly be sensible. Since it could imply that the doctor never treats the impatient, (1) is semantically ambiguous. The categorial disambiguator assumes no ambiguities of sense. This *might* be seen as a weakness in the approach. In a system that includes a full-fledged discourse component, however, the disambiguator could be made sensitive to discourse cues indicating an adjectival interpretation for such cases.

Perhaps the most important source of the fallibility of categorial disambiguation is the fact that no exhaustive set of disambiguation rules exists. Categorial disambiguation relies on an as yet unarticulated theory of category combination constraints. This being the case, the disambiguator has developed incrementally, through practice.

First, an initial set of the most intuitively obvious pattern-action clauses was constructed for the most common categorial ambiguities. Whenever sufficient constraint was unavailable using these rules, and whenever the disambiguator selected a category incorrectly, users were asked to select the appropriate category. These selections triggered a background mail process that reported the sentence, the particular ambiguity, and the user's selection to the implementor. This information proved indispensible in developing a large set of rules covering a broad range of ambiguous conditions.

## 5 A Categorial Disambiguation Rule

One simple categorial disambiguation rule decides whether a word (in this case only the word "to") is a preposition or the infinitival complement. It is presented schematically in Table 1.

| Condition | Consequent |
|---|---|
| 1. unambiguous noun phrase follows | preposition |
| 2. unambiguous verb phrase follows | complement |
| 3. adverb follows with following verb phrase | complement |
| 4. WH clause termination point | preposition |
| 5. non-WH clause termination point | complement |
| 6. next word is either a singular noun or a verb | complement |
| 7. next word is either a plural noun or a verb | preposition |
| 8. disambiguation failure | alternatives |

Table 1: Rule for disambiguating "to"

Each condition evaluates sequentially until one succeeds. The correct answer (rule consequent) replaces the alternatives in the category lists. Conditions 1 and 2 handle the simplest cases, in which the following categories unambiguously indicate a verb phrase or a noun phrase. Condition 3 handles split infinitives. Conditions 4 and 5 handle cases in which the word is at the end of a sentence or clause. If the clause begins with a subset of WH complements (who, which, what, where), "to" is considered to be a preposition (something up with which Winston Churchill would not put). Otherwise, "to" is made the infinitival complement (with an ellipsis). Conditions 6 and 7 handles cases in which the next element is also ambiguous, but is either a noun or a verb. If the next word would be a singular noun (e.g., "to store"), "to" must be a complement. If it would be a plural noun (e.g., "to stores"), "to" must be a preposition.

Condition 8 is the failure condition. The undisambiguated alternatives are returned. Rule failures always return the ambiguity. In such cases, succeeding disambiguations may propagate

enough constraint so that a reapplication of the rule would successfully resolve the ambiguity. Only when no possible additional sources of constraint exist will an ambiguity be considered irresolvable. In such cases, the user is asked to resolve the ambiguity, and the user's action is recorded so that the rule might be extended.

## 6 Heuristics for Rule Construction

The disambiguation rules are designed to minimize the amount of computation needed for successful resolution. Rules for ambiguities involving two alternatives attempt to rule in the correct category. Generally, for two alternative rules, conditions which are least computationally expensive to evaluate are evaluated first, while more expensive conditions are evaluated later.

Rules for ambiguities involving more than two alternatives attempt to rule out, rather than rule in, particular alternatives. For efficiency, alternatives considered least likely or easiest to rule out are checked first. If an alternative can be ruled out, control passes to the rule that disambiguates the remaining alternatives, and so on, until the ambiguity is resolved. Ruled-out alternatives do not reappear on the list of alternatives for a particular word in a particular sentence. Instead, only the remaining alternatives appear for possible later disambiguation.

## 7 Assessment of the Current Model

Categorial ambiguities are currently resolved prior to the detection of phrasal boundaries. The speedy development of a disambiguator with a broad range of coverage motivated the choice to implement it as a module separate from other parser components.

There is no reason, in principle, why the disambiguator could not be more tightly integrated with the parser. To do this, however, a stack of previously parsed words and categories would need to be maintained. Information regarding their linear order might be lost in the phrase structures. Also, since decisions regarding the appropriateness of particular parse rules often depend upon knowledge of succeeding categories, an integrated implementation would require occasional resolutions of categorial ambiguities ahead in the sentence.

As is well-known from expert-systems research, changes in one rule can sometimes produce unexpected negative results from other rules that were not changed. The categorial disambiguator is by no means free of these problems. They are largely eliminated, however, by (a) avoiding side-effecting consequents in the rules even when a successful condition might warrant the immediate disambiguation of a category elsewhere in the sentence, relying instead on constraint propagation, and (b) the installation of debugging tools that trace the evaluation of rules and rule conditions. As a result, the causes of failed disambiguations are simple to detect and thus also to remedy.

Variations on the current approach are certainly possible and might constitute interesting lines of research. For example, one might implement an expert-system variant, in which disambiguation rules are represented declaratively and perhaps weighted to indicate their relative values evidential impact upon a resolution. Such an approach would undoubtedly run slower than the prioritized procedural approach described here, but the model would also be more easily manipulable and applicable to other languages.

One might also attempt to remove the implementor from the debugging loop, implementing a variant which automatically learned new rules from disambiguation failures. The practical difficulty of this approach, however, involves the automatic detection of the reasons for such a failue. Errors might be due, for instance, to the specific subcategories of particular words. Without a full-blown theory of the role of subcategories in categorial disambiguation, it is difficult to see how a program could be made to select the correct subcategory consistently.

## 8 Related Work

Very little research seems to have been conducted on the resolution of categorial ambiguity. This has been somewhat surprising, since the technique is quite straightforward and the results are most powerful.

### 8.1 Wilks' Preference Semantics

As part of his "preference semantics" approach, Wilks [6] resolves categorial ambiguities by characterizing sentences as alternative sequences of semantic primitives and testing each for goodness of fit using a database of templates expressed in those primitives. To use one of Wilks' examples, sentence (3), in which the term "father" is categorially ambiguous, may be characterized as two alternative sequences of semantic primitives, (3a) and (3b).

| | | | | | | |
|---|---|---|---|---|---|---|
| (3) | Small | men | sometimes | father | big | sons. |
| (3a) | KIND | MAN | HOW | MAN | KIND | MAN |
| (3b) | KIND | MAN | HOW | CAUSE | KIND | MAN |

The alternative interpretations are then reduced to a sequence of "bare templates" by stripping off the adjectival KIND and the adverbial HOW, resulting in MAN MAN MAN and MAN CAUSE MAN. These two sequences are then matched against a database of legitimate bare templates. Since only the second sequence appears in that database, (3b) is correctly chosen as the appropriate interpretation.

It is important to note that Wilks' technique was not designed to resolve categorial ambiguities. It does so as a by-product of its resolution of sense ambiguities. This does not mean that sense disambiguation, by this method or by any other (e.g., [3]) obviates categorial disambiguation. Quite the contrary, it means that categorial disambiguation can be an inexpensive prelude to sense disambiguation. Had sentence (3) been categorially disambiguated earlier, "father" would have been recognized as a verb, so the nominal usage of "father" would already have been ruled out. No reduction to primitives and no matching in a template database would have been needed.

### 8.2 Breadth-First Parsing

Another alternative method for resolving categorial ambiguities – pursuing all categorially possible parse paths in breadth-first fashion [2] – is exponential in the number of categorial ambiguities. The number of alternative parse paths which would be tried is the product of all possible category assignments. For sentence (1), for example, the number of alternative paths would be:

**The doctor might cure the patient**

$$1 \times 2 \times 2 \times 2 \times 1 \times 2 = 16$$

Worse, many categorial ambiguities involve three or four possible category assignments. For example, some words can be progressive verb forms ("You are winning"), gerundive adjectives ("the

winning entry"), or gerundive nouns ("Winning is everything"). Extremely common words, like "in", take even more possible category assignments.

### 8.3 Phenomenologically-Plausible Parsing

Waltz and Pollack [5] present a connectionist model in which categories are disambiguated concurrently with other lexical ambiguities. Unlike the serial processing case, prior resolution of categorial ambiguities in a connectionist models would not result in any significant time savings. In fact, in a connectionist model, sequential resolution of categorial ambiguities prior to the resolution of other ambiguities would consume more time. There is a trade-off, however. As in the serial case, earlier categorial disambiguation would constrain the range of possible sense interpretations. Instead of saving time, in the case of massively parallel processing, prior categorial disambiguation would conserve processors.

## 9 Conclusions

Categorial disambiguation is a computationally inexpensive means for reducing the ambiguity of a sentence. While categorial disambiguation does not resolve all the ambiguities that may appear in a sentence (e.g., anaphor ambiguities and ambiguities of prepositional attachment, clausal attachment, and sense), it does eliminate a source of ambiguity which pervades sentences. Moreover, the prior resolution of categorial ambiguities radically simplifies procedures that resolve these other classes of ambiguity.

The implemented disambiguator remains in a development stage. No tests have yet been performed, using representative texts, to estimate its degree of coverage. Nevertheless, experience with moderately complex sentences indicates that the current set of rules is quite robust. As disambiguation failures are detected, the rulebase is extended, making it more robust. Most importantly, its efficiency over breadth-first approaches and the leverage it provides for other forms of disambiguation suffice to warrant use and extension of the technique.

As a final note, a large amount of syntactic knowledge is embedded within the categorial disambiguation rules. These rules are also a potential source of considerable constraint for the resolution of ambiguous morpheme sequences output by a speech recognition program. By applying a categorial disambiguator to this output, syntactic constraints on possible morpheme sequences may be applied without the overhead involved in testing alternative parse-tree constructions.

## 10 Acknowledgments

This paper was improved by comments from John Batali, J. Michael Brady, John C. Mallery, Sidney Markowitz, and two anonymous reviewers. Erik Devereux helped typographically. The author remains responsible for the content. Much of the research reported here was conducted at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology.

## 11 References

[1] Duffy, G., "Viewing Parsing as Patterns of Passing Messages," forthcoming, 1986, available from author.

[2] Martin, W. A., Church, K. W., and Patil, R. S., "Preliminary Analysis of a Breadth-First Parsing Algorithm: Theoretical and Experimental Results," TR 261, MIT Laboratory for Computer Science, 1981.

[3] Small, S., "Word Expert Parsing: A Theory of Distributed Word-Based Natural Language Understanding," TR 954, University of Maryland, Department of Computer Science, 1980.

[4] Waltz, D. L., "Understanding Line Drawings of Scenes with Shadows", in Patrick H. Winston, ed., *The Psychology of Computer Vision*, New York, McGraw-Hill, 1975. pp. 19-91.

[5] Waltz, D. L., and Pollack, J. B., "Phenomenologically Plausible Parsing", *AAAI-84: Proc. of the Nat. Conf. on Artificial Intelligence*, AAAI, August 1984, pp. 335-339.

[6] Wilks, Y., "Preference Semantics," Memo AIM-206, Stanford Artificial Intelligence Laboratory, Stanford California, 1973.