PARSING AGAINST LEXICAL AMBIGUITY

Rob Milne
Dept of Artificial Intelligence and School of Epistemics
University of Edinburgh
Edinburgh, Scotland
EH8 9NW

ABSTRACT

Marcus' original deterministic parsing included almost no part-of-speech ambiguity. In this paper, the addition of part-of-speech ambiguity to a deterministic parser written in Prolog is described. To handle this ambiguity, it was necessary to add no special mechanisms to the parser. Instead the grammar rules were made to enforce agreement, and reject ungrammatical sentences. The resulting system is very effective and covers many examples of ambiguity.

INTRODUCTION

Most words can be more than one part of speech. For example, many words that can be a noun, can also be a verb, many -ing verbs can also act as adjectives, many prepositions can serve as particles, several modals can also be nouns, and some relative pronouns can also be determiners. In order to analyze a sentence, it is necessary to decide which part of speech a given word is in the sentence. Deciding which part of speech a word is during sentence processing shall be refered to as Lexical Ambiguity(LA). If a parser is to handle a wide range of English and ambiguity, it is necessary for it to handle this problem.

STATE OF THE ART

Marcus [1977] showed that a wide range of English grammar could be parsed determinsitcally, that is without every making a mistake and having to backtrack. But in Marcus' parser, almost every word was defined as only one part of speech. For example in his parser, "block" could only be a noun, making the following sentence unacceptable to the parser.

[1] Block the road.

With so little ambiguity, it is not surprising that Marcus's parser could work deterministically. For determinstic parsing to be a serious claim, it must be shown that it is possible to parse determinstically sentences which contain part-of-speech ambiguity. Is deterministic parsing still possible when part of speech ambiguity is included?

The answer to this question can be thought of as the first major test for determinsitic parsing. If it is able to handle part-of-speech ambiguity easily, this will be a major reinforcement of the deterministic parsing strategy. If it cannnot handle LA, the theory will collapse.

The first approach to LA for a deterministic parser was [Milne 78]. This work dealt solely with noun/verb ambiguity. When a noun/verb word was discovered, a special packet of rules was activated to decide which part-of-speech the word should be. For example, a typical rule stated that "to" followed by a noun/verb word meant that the noun/verb word was being used as a verb, and would disambiguate it as such. The rest of the grammar dealt with the disambiguated word.

Although this approach was very effective, the rules were very special case, and many rules would be needed to handle all the possibilities.

THE DEFAULT CASE

I have implemented a deterministic parser in Prolog [Pereira 78] similar to Marcus' but extended it to allow words to be defined as multiple parts of speech. The parser has appoximately 80% of Marcus' original grammar, but the grammar has been extended to cover the domain of mechanics problems. (MECHO) [Bundy 79a,79b].

To extend the Prolog parser, each word in the dictionary was syntactically defined as all parts-of-speech it could function as, given the grammar. The only other initial modification necessary was to alter the attach function to disambigute the word to the part-of-speech it is being attached as. For example if "block" is attached as a noun, it will be disambiguated to a noun. Because of the expectations of the parser, represented by the packets, and the constraints of neighboring items, represented by the buffer pattern matching, a large number of cases were handled without further modification.

For example in the sentence:

[2] The block is red.

The parser will be expecting a noun after

the determiner, and hence only the rules for nouns in nounphrases will be active. "Block" will be used as a noun, and the verb usage never considered.

Similary in the case:

[3] Block the road.

The rule for Imperative at the sentence start will match off the verb features of "block", and the noun usage will not be considered.

The current parser can handle the following examples with no special rules:

noun/verb
[4] The block will block the road.
[5] I want to block her.
[6] The pot cover screws tightly.
pronoun/poss-det
[7] Tom hit her.
[8] Tom hit her dog.
noun/modal
[9] The trash can be smelly.
[10] The trash can is smelly.

THE DIAGNOSTICS

Marcus allowed several "function" words to be more than one part of speech. For example "have" could be an auxverb or a main verb, "that" could be a comp, determiner, or pronoun, and "to" could be a preposition or a auxverb. To handle these ambiguities, Marcus had a "Diagnostic rule" for each word. The diagnostic rules matched when the word it was to "diagnose" arrived in the first buffer, and used the 3 buffer look ahead to resolve the ambiguity. Each Diagnostic rule could ask questions concerning the grammatical features of the contents of the 3 buffers, as well as the partial item being built. As a result these rules were very complex and cumberson compared with the rest of the rules. But these rules seemed necessary to preserve the generality of the other rules.

For example, the "HAVE-DIAG" decided if the sentence was a Yes-No-Question(YNQ) or an Imperative, and hence "have" a main verb or auxverb. The rule was as follows:

```
[have][np][verb] ->
If 2nd is noun singular,n3p
or 3rd is not +en then run Imperative.
        else run Yes-No-Question.
```

and decided between:

```
[auxverb][np] -> Yes-No-Question
[tnsless verb] -> Imperative
```

at the start of the sentence.

To alter the YNQ rule for the special case of "have", would ruin the simple generality of the rule, and lose the linguistic generalization it captures.

But the Marcus Parser assumed it would only be given grammatical sentences. If the Marcus parser was given an ungrammatical sentence, it might pass it as legal. For example the parser would pass as legal:

[11] *Is the boys running
[12] *Is the boy run?

Notice they both match the YNQ pattern.

Clearly for the rule YNQ to run, the auxverb must agree in number with the subject, and in affix with the verb. If we modify the YNQ rule to enforce this agreement, then only [13] will match the YNQ rule:

[13] Have the boys taken the exam?
[14] Have the boy taken the exam.
[15] Have the boys take the exam.
[16] ?Have the boy taken the exam.

In fact, if we enforce agreement on the YNQ rule, it will perform exactly the same as the old HAVE-DIAGNOSTIC, and the diagnostic is made redundant.

Closer inspection of the diagnostics and the grammar rules they decide between, reveals that the grammar rules will in general pass ungrammatical sentences as legal. If these rules are then corrected, using agreement and grammaticallity, then all the diagnostics are made redundant and no longer needed.

In order to handle part-of-speech ambiguity in a determinsitic way, the parser does not need special "Diagnostice rules". If the grammar enforces agreement, and rejects ungrammatical strings then ambiguity handling happens automatically.

THE THAT-DIAGNOSTIC

The most complicated of all the diagnostics, was the THAT-DIAGNOSTIC. This rule decided if "that" was a determiner, pronoun, or a comp. In Marcus' parser, 3 rules were needed for this decision. Also, if Marcus' diagnostic decided that "that" was to be a determiner, then it would be attached after the nounphrase it would be a determiner for, was built! In Church [1980], the THAT-DIAGNOSTIC is only one rule, but extremely complicated. His deterministic parser can handle the widest range of "that" examples, but the diagnostic is seemingly the most complicated in the grammar.

Following the above methodology though, the diagnostic can be made redundant. "that" can only be a determiner if the word following it

will take a determiner. In Marcus' original parser, the rule DETERMINER made no check for grammaticallity, and would attempt to parse the following fragements:

[17] *the the boy
[18] *the he
[19] *the tom
[20] *a blocks

If the rule DETERMINER is fixed to reject these examples, then the determiner usages will all work properly. Similary, the rule PRONOUN would pass ungrammatical strings, so this was altered. Finally, only the comp use of "that" are left, and the parser's normal rules can handle this case. By simply altering the above rules to reject ungrammatical strings, the following sentences can be parsed with no special diagnostic additions to the parser.:

[21] I know that.
[22] I know that boy.
[23] I know that boy hit mary.
[24] I know that was nice.
[25] I know that that was nice.
[26] I know that he hit mary.

GARDEN PATHS

After altering the grammar, so there were no special rules for ambiguity, the following sentences were still a problem:

[27] What little fish eat is worms.
[28] That deer ate everything in my garden surprised me.
[29] The horse raced past the barn fell.
[30] The building blocks the sun faded were red.

But for each of these, there is a partner sentences, showing these ae potential garden paths [Milne 1980b].

[31] What little fish eat worms.
[32] That deer ate everything in my garden.
[33] The horse raced past the barn.
[34] The building blocks the sun.

As Marcus stated in his thesis, a deterministic parser cannot handle correctly a garden path sentence. But people also fail on garden path sentences. Since deterministic parsing should model human performance, and not exceed it, it is acceptable for the parser to fail. Instead these potential garden path situations are resolved using semantic information [Milne 1980b].

Enforcing number agreement fails when a word is morphologically ambiguous. This problem has not been examined yet.

FREE TEXT

A simulation of these rules was conducted by hand on an article in TIME [1978] and the front page of the NEW YORK TIMES [1978]. The parser's rules disambigution was correct for 99% of the occurances that the grammar could cover. (some ambiguities are not yet handled).

A POSSIBLE EXPLANATION

At first glance, English looks extremely ambigous and the ambiguity very difficult to handle. But given the constraints of grammaticallity, most of the ambiguity disappears. For only one of the possible multiple choices will generally be grammatical. People do not seem aware of all the ambiguity in the sentences they process (excluding global ambiguity examples). This and the paper suggests that handling ambiguity causes no additional load on a parser, a very desirable and intuitively acceptable result. In other words, grammaticallity and LA handling are directly related.

CONCLUSION

In this paper, I have described adding part-of-speech ambiguity to a version of the Marcus determinstic parser. The only additions necessary to the parser, were having the attachment function coerce the words to the part of speech the word is attached as and the grammar had to be altered so the rules would reject ungrammatical sentences, and made to enforce number and affix agreement. With these additions, the parser is able to handle a very wide range of ambiguity, with no special rules, and no need to backtrack. The resulting lexical ambiguity handling is very flexible and has a high success rate when simulated on free text.

This work is far from complete. In this paper we have not discussed syntax/semantics interaction and global ambiguity. For comments on these, see [Milne 1980].

ACKNOWLEDGEMENTS

Examples [13,14,27,28] are from Marcus.
Examples [21-26] are from Church.
This paper describes work done under an Edinburgh University Studentship.

BIBLIOGRAPHY

Bundy, A., Byrd, L., Luger, G., Mellish, C., Milne, R.,Palmer, M. [1979a] "MECHO: A Program To Solve Mechanics Problems",DAI Working Paper No. 50.

Bundy, A., Byrd, L., Luger, G., Mellish, P., Palmer, M. [1979b] "Solving Mechanics Problems Using Meta-Level Inference", IJCAI-79

Church, K. [1980] "On Memory Limitations in Natural Language Processing", unpublished MSc. Thesis, MIT AI LAB.

Marcus, M. P. [1977] "A Theory of Syntactic Recognition for Natural Language", unpublished Ph.D. thesis, MIT.

Milne, R. [1978] "Handling Lexical Ambiguity in a Deterministic Parsing Enviornment", unpublished B.Sc. thesis, MIT.

Milne, R. [1980a] "A Framework for Deterministic Parsing Using Syntax and Semantics", DAI Working Paper No. 64.

Milne, R. [1980b] "Using Determinism to Predict Garden Paths",AISB 80 Conference Proceedings.

Pereira, C.M., Pereira, F.C.N. and Warren, D.H.D. [1978] "User's Guide to DECsystem-10 PROLOG", Available from the AI Dept, Edinburgh.

The New York Times Wednesday, April 28, 1978 Vol CXXVII No. 43922

TIME [Jan. 9, 1978] Good ole Burt; Cool-eyed Clint