Parsing and Recovery

Vincenzo Lombardo
Dipartimento di Informatica - Universita` di Torino
c.so Svizzera, 185 - 10149 Torino - Italy
Centro di Scienza Cognitiva - Universita` di Torino
via Lagrange, 3 - 10123 Torino - Italy
vincenzo@di.unito.it

Abstract

The paper introduces a general model of recovery from errors in parsing. The mechanism proposed returns selectively on the choice points, in order to identify the one that was badly resolved, and could have caused the error. Then, it selects an alternative previously discarded and finally selectively repairs the appropriate fragments between the ambiguous region and the breakdown region. Both the psycholinguistic and the computational features of the model are put in evidence.

Introduction

The resolution of syntactic ambiguities has received much attention in the literature. Attachment preferences have been devised with both the goals of explaining the behavior of the human parser (Kimball, 1973; Frazier & Fodor, 1978; Ford, Bresnan & Kaplan, 1982) and equipping a system with an efficient mechanism for the selection of a promising syntactic structure (Shieber, 1983; Hobbs & Bear, 1990; Huyck & Lytinen, 1993); part-of-speech disambiguation has been dealt with in the context of deterministic parsers (Milne, 1986) and automatic text tagging (Church, 1988; DeRose, 1988; Hindle, 1989).

So far, cognitive models have made much progress in accounting for the initial preferences in human parsing and in characterizing garden-path sentences; automatic systems have improved the capability of reducing the size of the search space by selecting plausible structures according to cognitive and computational criteria and restricted domain knowledge. Path selection tries to optimize local operations, when only the current input phrase and some portions of syntactic (in incremental processing also semantic and contextual) structures are accessible.

However, the best local choice can turn out to be wrong. Most models of human parsing roughly distinguish two classes of failures, irrecoverable garden-paths and simple locally revisable sentences; most system architectures include bookkeeping mechanisms that avoid blind backtracking. Computational models of recovery are very rare in the literature.

The goal of this paper is to introduce a general model of recovery that takes cognitive insights as a starting point. The model has been tested on a set of heuristics that recover some well-known cases of breakdown, reported in the literature on syntactic ambiguity and garden path theory.

The organization of the paper is as follows. The next section introduces the parsing process, in order to understand what situation the system encounters at the breakdown point, in terms of active structures and abandoned paths. The recovery mechanism is described in the third section, by considering the specific phases that compose the global process. Finally, some conclusion and comments on the related work are reported. The appendix reports a small subset of the sentences used for testing.

The Parsing Algorithm

The parsing algorithm goes left-to-right, pursuing a selective top-down strategy that joins the predictive power of top-down parsers and the bottom-up filter provided by scanning the category of the word in input. This strategy accounts for two aspects of the human processor: on one hand, the facility of committing syntactic expectations, on the other, the data-driven triggering of such a facility. At the points of non-determinism, the parser pursues a limited form of parallelism, in order to evaluate the best path to follow. In this phase the parser takes advantage of the notion of compact representation (subtree sharing and local packing (Tomita, 1987)) in order to avoid duplicating the same structure on several paths (fig. 1a). Some parts of the structure cannot be shared by all the paths: for instance, the three edges NP-PP, VP-PP, S-PP, in fig. 1a are mutually exclusive. To signal that some elements cannot occur together in the same structure, the processor labels them with indices: an index has the form i.j, and identifies the j-th alternative for the i-th point of non determinism. Elements labelled i.j cannot cooccur with elements labelled i.k, $k \neq j$. Given the top-down character of the parsing process, an index associated with an element is intended to label the whole subtree rooted in it. The parts of the structure that do not depend on a particular choice are left unlabelled and are meant to belong to the structure whatever path is followed. A *path* is precisely defined as an exhaustive set of indices that can cooccur, where exhaustive means that the set contains at least one representative of each ambiguity encountered, and, because of cooccurrence, it contains exactly one. The structure that corresponds to a path is one possible *parse*.

The path to follow results from an evaluation, in terms of syntactic preference and semantic consistency, of the various partial structures built in the parallel phase. One parse, called the *active parse* and corresponding to a specific path, is selected for continuation. The remaining partial structures are not deleted, even if not reachable in the currently active path, and are possibly reconsidered in the recovery phase.

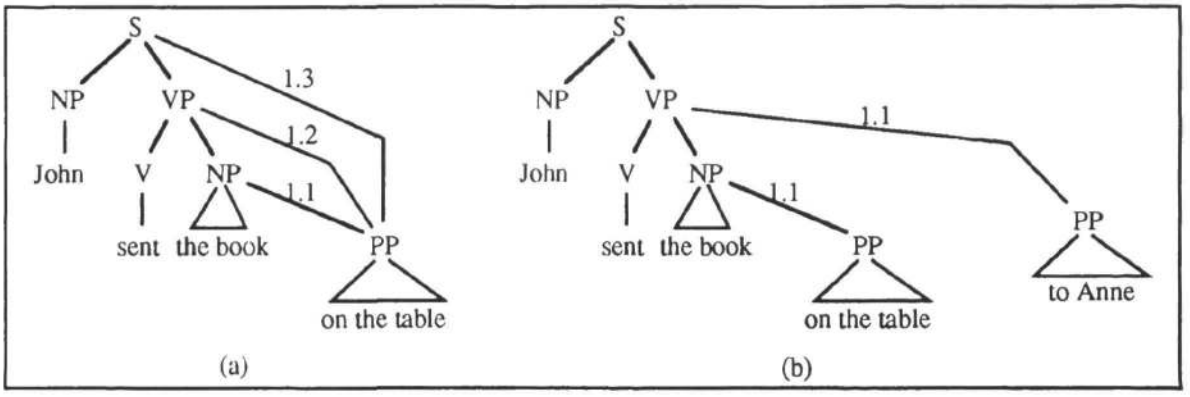

Figure 1. Two snapshots in parsing: the structures are simplified to point out the index mechanism.

Let us consider the analysis of the sentence "John sent the book on the table to Anne" (see fig. 1). The attachment of "on the table" is ambiguous (fig. 1a) and the three alternatives are labelled 1.j, j∈ {1, 2, 3}. The three possibilities are mutually exclusive and the structure in fig. 1a actually represents three partial parses, each including just one of the attachments.

Attachment preferences lead to a preferred structure (the alternative which is preferred in fig.1b (1.1) corresponds to Late Closure). Finally, the last PP is attached to VP without ambiguity, because of the subcategorization constraints given by "sent" and the preferred attachment of a PP headed by "to" to an *action* (sent) rather than an *object*. (table, book). The index 1.1 also labels the new attachment because VP has been made reachable after the selection of 1.1 at the previous non-deterministic point (compare with selecting 1.2 or 1.3).

Recovery

Psycholinguistic literature has not paid much attention to the problem of recovery, even if a more or less explicit mechanism has been widely conjectured, given the garden path phenomena and the general limitations of the working memory[1]. One of the results of the psycholinguistic research has been the selective reanalysis hypothesis (Frazier & Rayner, 1982), which suggests the existence of a human capability to "quickly identify the source of an erroneous analysis of temporarily ambiguous material".

In terms of a computational model, an "intelligent" recovery mechanism should not try out each choice point (like a standard backtracking procedure), but should carry on a selective search. The expectations that remain unsatisfied in the currently active parse and the features carried by the input material that caused the breakdown (included a premature end of sentence) form a body of information that guides a heuristic-based search of choice points. Once the wrong choice is identified, the same information contributes to select an alternative choice among those previously discarded. Finally, the active parse is repaired complying with the path change.

[1]Some exceptions to this claiming are summarized in section 5.

The recovery mechanism presented in this paper takes the selective reanalysis hypothesis as a starting point. The model (fig. 2), that has been equipped with a set of heuristics that account for common breakdowns reported in the literature, consists of three phases: error diagnosis, that translates the information available at the breakdown point into a unique symbol; selection of an active index i.j and proposal of an alternative index i.k (k≠j), given the error diagnosed and the active structure; repair of the elements of the representation labelled i.j. These phases are described in detail in the following sections.

Diagnosis

The output of the diagnostic module is an error type, a symbol that conveys the useful information available at the breakdown point. The taxonomy of the possible errors is shown in fig. 3. There are two major error classes: *Extra*, that indicates linguistic material in excess with respect to the expectations, and *Missing*, that indicates that an element that was expected is "missing". The error is more informative if it is possible to identify the syntactic category of a phrase in excess (e.g. Extra-NP) or expected (e.g. Missing-Verb) or the grammatical relation deducible from the surface form of a phrase (e.g. *she* Extra-Subj) or predicted by the currently active subcategorization frame (e.g. Missing-Obj). The low levels of the taxonomy (not in the figure) contain very specific types that account for further syntactic features, like gender, number, finiteness of verbs, ...

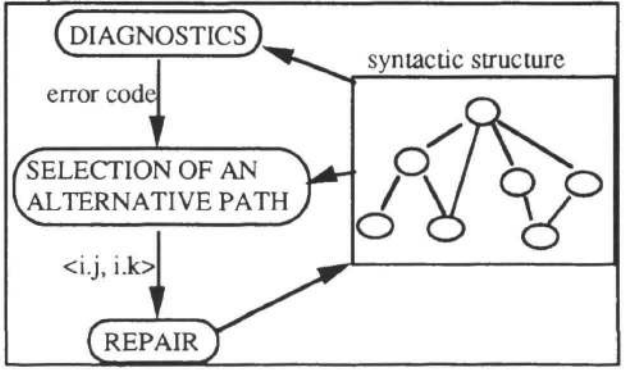

Figure 2. The recovery model.

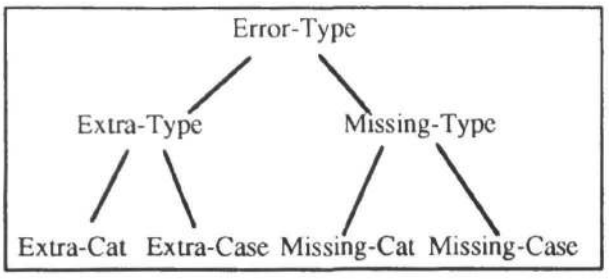

Figure 3. Taxonomy of error types: the leaves are omitted (they depend on the specific grammar and subcategorization frames).

Here are some examples of garden paths, that if interpreted in an unbiased context originate the error types in brackets:

1) The horse raced past the barn *fell* (Bever, 1970) (*Extra-Verb*)
2) The prime number few ◊ (Milne, 1982) (*Missing-Verb*)
3) That information is important *is* doubtful (Tomita, 1987) (*Extra-Verb*)
4) I gave her food for the dog ◊ (*Missing-Recipient*)

The breakdown point is indicated by a word in italics or the symbol ◊ (end of sentence); the ambiguity point, where the wrong choice was made, is underlined.

The diagnostic module is a set of rules that take into account the situation at the moment of the breakdown, in terms of the features of the input word and of structures built, and produce an error symbol. Most breakdowns are covered by two very intuitive rules, namely

R1) IF *the input phrase is of category Cat*
THEN *return Extra-Cat*

R2) Let N be a node
IF *end-of-sentence &*
a phrase of category Cat, expected as a dependent of N, is missing
THEN *return MissingCat*

Specific rules diagnose more informative error symbols:

R3) IF *the input word is a finite verb &*
no subject for it has been found yet
THEN *return Missing-Subj*

Each error symbol covers a set of breakdown phenomena: error symbols are determined empirically and their number depends on the size of the grammar and the expertise of a system in recovery. A subset of the sentences used for testing the model is displayed in the appendix, together with the error code detected.

Selection of an alternative path

The goal of this phase is to identify an active choice i.j that led to the breakdown. Each error code has a search procedure associated, that selects in the *active parse* an element that, according to some heuristic, could be "related" to the error code. If this element is labelled with an index (i.j), this means that its construction was the result of a preference matter rather than a deterministic process, and, then, it could be a candidate for revision. An alternative index i.k, that satisfies some further conditions, is returned. The search procedures incorporate some rationale underlying the error type. For instance, in the case of *Extra-verb*, it could be guessed (see S1 below) that some word encountered was incorrectly parsed as the (current) verb, thus preventing the correct parsing of the current input word ("raced" in ex. 1). Here are some examples:

Extra-Verb

S1) *Search for a verbal node labelled with an index i.j*
-> Return i.k such that introduces a verb expectation

S2) *Search a subcategorization frame curretly active such that:*
1. it is labelled with i.j
2. the corresponding verb has an alternative subcategorization {labelled i.k} that contains a clausal complement
-> Return i.k

Extra-Subj

S3) *Search for the current SUBJ, say NP_m, labelled with i.j*
--> Return i.k such that NP_m is not SUBJ

Missing-Subj

S4) *Search for the NP immediately preceding the input verb such that its attachment is labelled with an index i.j*
--> Return i.k such that NP is SUBJ

The refinement of the error types guarantees the non-overlapping of the applicability conditions for the search procedures: at the moment, the testing on the search procedures have produced incouraging results, since no conflicts arose.

Repair

The last phase involves the adjustment of the syntactic structure according to the new active path given by the previous path where i.k has replaced i.j. The first step is to include in the active parse the element E, labelled i.k, that was selected in the previous phase. Then, repair computes the new nodes suitable for expansion and starts a forward processing that is very similar to the parsing algorithm outlined above with some differences. The aim is to save computation time by reusing those structures that are compatible with the new active path, because not dependent on the choice indexed i.j that revealed to be wrong.

The repairing phase is described by the algorithm in fig. 4. The repairing phase (see fig. 4) "re-parses" the words comprised between the ambiguous region (BEGIN), just repaired with the new selection, and the disambiguating region (END), where the breakdown occurred. "Re-parsing" is very similar to the syntactic analysis illustrated in section 2 with one major difference: if the word in input is already parsed in a structure S that is still active after the change of path (it does not depend on the wrong choice), then only the attachment to the new active structure is recomputed and, in case of ambiguity, the best attachment of S is evaluated and the "re-parsing" jumps to the input word that comes after S; otherwise w is parsed in the usual way. When the "re-

parsing" arrives at the input word that caused the breakdown, the parsing process is restarted to finish the analysis.

```
set BEGIN to the rightmost word in the
    ambiguous region selected in the phase 2
set END to the the current input word
for each word w from BEGIN to END do
    compute the set of reachable nodes
    if w already belongs to a structure S that is
                                active in the new path
    then
        compute the possible attachments of the
                structure S to the reachable nodes
        select the best attachment
        jump to the first input word that follows
                                the structure S
    else parse w
```

Figure 4. The repairing algorithm.

An Example

The syntactic structure adopted in our implementation is a dependency tree (see fig. 5), a set of binary head-modifiers relations over the words, that represent the predicate-argument structure of the sentence.[2] The top-down left-to-right parsing algorithm "creates" nodes of a certain syntactic category and then "fills" them with one input word. The correct parse of the example sentence, "Though Hilda finally agreed to sing the songs she chose turned out to be just awful", is represented in fig. 5 (in the following figures, for the sake of simplicity, we will omit the relation labels on the arcs).

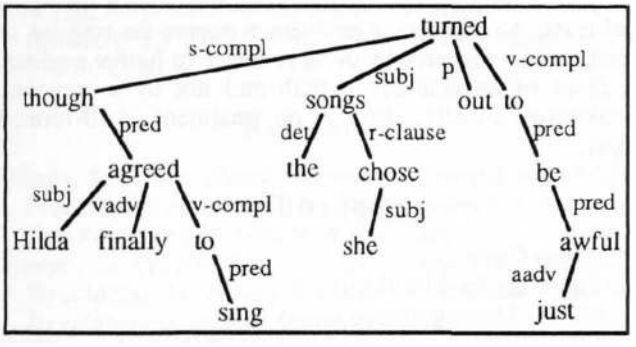

Figure 5. Dependency tree of the sentence **Though Hilda finally agreed to sing the songs she chose turned out to be just awful**. The word order is given by the orientation of the arcs. Labels on arcs represent the functions of modifiers, and are used in the semantic interpretation.

[2] The respective merits of constituency and dependency approaches as well as the relative mathematical power are discussed in a number of publications. The choice of this structure does not affect the generality of the recovery model.

After the unambiguous parsing of "Though Hilda finally agreed" as a sentential complement of a "yet-to-come" main verb (empty left side of node V1 in fig. 6), the attachment of "to sing" is competed between an Early Closure of "agreed", as would be required by

Though Hilda finally agreed(,) to sing those songs she ought to be a soprano

and a Late Closure, as in

Though Hilda finally agreed to sing(,) the songs she chose turned out to be just awful

giving the attachment labelled 1.2 and 1.1 respectively (fig. 6). If we follow the LC preference (1.1), the attachment for "the songs" is ambiguous between an EC and a LC of "sing", as exemplified by the two possible continuations

Though Hilda finally agreed to sing(,) the songs she chose turned out to be just awful

Though Hilda finally agreed to sing the songs(,) she chose very awful tunes

If we prefer again LC (2.1 in fig. 6), we have two attachments for "she", namely "sing" (LC) and the main verb of the sentence which is yet to come, both acceptable in the global contexts given respectively by

Though Hilda finally agreed to sing(,) the songs she chose turned out to be just awful

and

Though Hilda finally agreed to sing the songs(,) she chose very awful tunes

The choice for LC gives the structure in fig. 6, where the currently active path is given by the set {1.1, 2.1, 3.1}, and the abandoned choices are represented as dashed arcs. Notice that the node V1 is "empty" in the active path.

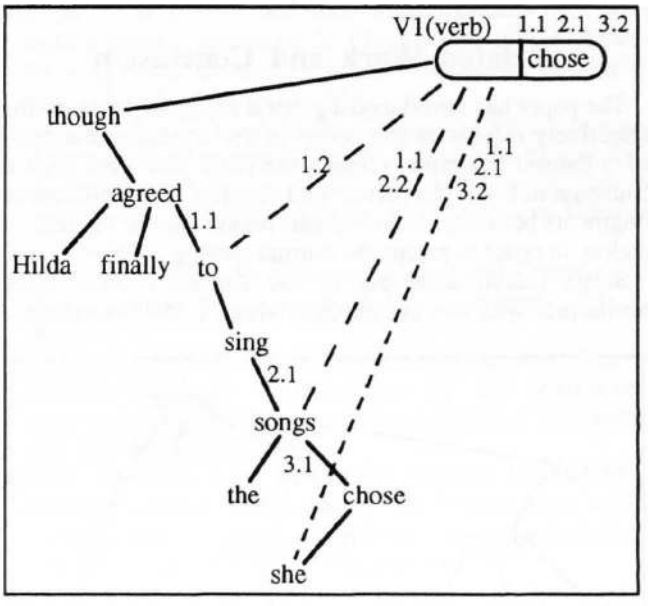

Figure 6. Syntactic structures after parsing "Though Hilda finally agreed to sing the songs she chose". The active arcs are given by plain lines, dashed lines are part of abandoned paths. The active path is given by the set of indices {1.1, 2.1, 3.1}. "chose" in V1 is not active, since its index 3.2 is not active.

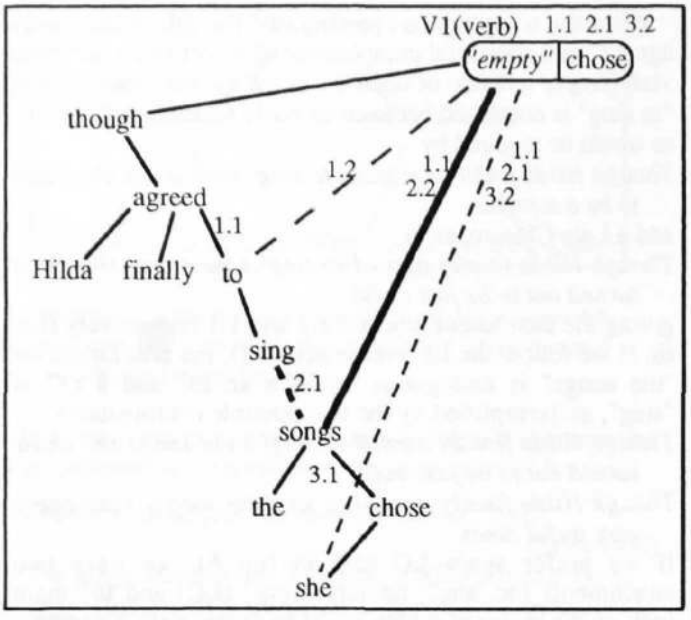

Figure 7.

When the word "turned" arrives, an error Missing-Subj is generated (rule R3) and the procedure S4 selects the NP rooted in "songs" and the index 2.1 for revision. The alternative choice 2.2 connects "songs" with V1 (fig. 7). The repairing phase inserts "turned" into the structure: the reachable nodes are "chose", "songs" and empty V1, but the only possible operation is to insert "turned" into V1. The rest of the input gives no problems and the final structure with the active indices is in fig. 8.

Related Work and Conclusion

The paper has introduced a general model of recovery that selectively returns to the points in the structure that could have caused the error, retracts the parts that were built in consequence of the error and repairs the appropriate fragments between the ambiguous region and the breakdown region, in order to restart the normal parsing.

Some recent attempts in the literature have some similarities with this mechanism. Abney (1993) introduces a

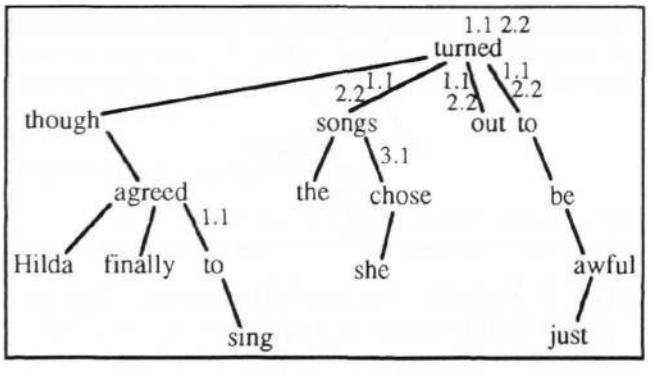

Figure 8. The complete dependency structure, given by the path {1.1, 2.2, 3.1}.

repairing mechanism into the incremental parser CASS. CASS works on tagged input text and the repairing filters propose new phrase structures in correspondence of specific error labels, that are particular non terminal symbols with productions associated. Some TMS-based proposals have tried to apply general techniques of reason maintenance to natural language: the JTMS approach (Zernik & Brown, 1988) has been criticized because of the impossibility to evaluate the best parsing route in presence of multiple alternatives; the ATMS approach (Charniak & Goldman, 1988), which overcomes this drawback by maintaining multiple interpretations in assumption-based contexts and switching between them, leads to a combinatorial explosion and is not able to perform default reasoning (this approach can be simplified by ATMS-styling a chart parser (Wiren, 1990)). In the field of cognitive modeling, Eiselt (1989) describes a recovery mechanism mostly applied to lexical ambiguity resolution in the context of a computational model called COMPERE (Mahesh & Eiselt, 1994); Stevenson (1994) has introduced a unified model for parsing and recovery that aims to account for the fine scalability of behaviors of human parsing, in dependency of recency factors.

In psycholinguistic terms, the model takes into account the expertise that human subjects own on recovery: an error label refers to a type of block that is recognizable by a human listener, since s/he already underwent and overcame the same difficulty in the past.

This paper is centered on syntactic ambiguities and the consequences derived from a wrong syntactic choice. However, the technique, which already accounts for errors in subcategorization, can be extended to cope with general semantic failures and erroneous pragmatic inferences. The heuristics introduced can be furtherly refined through the analysis of the larger set of examples and after a testing on real texts. An interesting problem concerns the revision of interpretation triggered by the appearance of further evidence in favor of an abandoned path and not by a complete breakdown. Finally, there is no treatment of ill-formed input.

Appendix

Missing-Case

a.1) I gave her food for the dog Δ
ERROR: Missing-Dative (give)
AMBIGUITY: ADJECTIVE /PRONOUN (her)
CHOICE: ADJECTIVE

a.2) I put the book in the bathroom Δ
ERROR: Missing-Destination(put)
AMBIGUITY: PP ATTACHMENT (NP/VERB)
CHOICE: NP ATTACHMENT

a.3) John told the girl that he had married *that* he never loved her
ERROR: Missing-Obj(marry)
AMBIGUITY: RELPRON/COMPLEMENTIZER (that)
CHOICE: COMPLEMENTIZER

a.4) Though George went on reading the story really *bothered* him
ERROR: Missing-Subj(bothered)

AMBIGUITY: LC/EC (reading)
CHOICE: LC

Extra-Case

b.1) He put the cattle in the barn *out* to pasture for spring grazing
ERROR: Extra-Destination(put)
AMBIGUITY: PP ATTACHMENT (NP/VERB)
CHOICE: VERB ATTACHMENT

b.2) He mailed the ticket to London *to Mary*
ERROR: Extra-Destination(mail)
AMBIGUITY: PP ATTACHMENT (NP/VERB)
CHOICE: VERB ATTACHMENT

b.3) I told the girl that you kissed *the story*
ERROR: Extra-Obj(tell)
AMBIGUITY: RELPRON/COMPLEMENTIZER (that)
CHOICE: COMPLEMENTIZER

b.4) I gave her food *to the dog*
ERROR: Extra-Dative(give)
AMBIGUITY: ADJECTIVE /PRONOUN (her)
CHOICE: PRONOUN

Missing-Verb

c.1) The prime number few Δ
AMBIGUITY: ADJECTIVE /NOUN (prime)
CHOICE: NOUN

c.2) The woman rushed to the hospital and *forgot* her laundry Δ
AMBIGUITY: FINITE /INFINITE (rushed)
CHOICE: INFINITE

Extra-Verb

d.1) The horse raced past the barn *fell*
AMBIGUITY: FINITE /INFINITE (raced)
CHOICE: FINITE

d.2) That information is important *is* doubtful
AMBIGUITY: DET/COMPLEMENTIZER (that)
CHOICE: DET

References

Abney, S. (1993). Rapid Incremental Parsing with Repair, *Proceedings of the 6th New OED Conference: Electronic Text Research* (pp. 1-9). Waterloo, Ontario.

Bever, T. (1970). The Cognitive Bias for Linguistic Structures. In Hayes J. (Ed.), *Cognition and the Development of Language* (pp. 279-352). New York: John Wiley And Sons.

Charniak, E. & Goldman, R. (1988). A Logic for Semantic Interpretation. In *Proceedings of 26th Annual Meeting of the Association for Computational Linguistics* (pp. 87-94). Buffalo.

Church, K. (1988). A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of the Second Conference of Applied Natural Language Processing* (pp. 136-143).

DeRose, S.J. (1988). Grammatical Category Disambiguation by Statistical Optimization. *Computational Linguistics*, 14, 31-39.

Eiselt, K. (1989). Inference Processing and Error Recovery in Sentence Understanding. (Tech. Report 89-24). Doctoral Dissertation. Irvine, CA: University of California.

Ford, M., Bresnan, J. & Kaplan, R. (1982). A competence-based theory of syntactic closure. In Bresnan J. (ed.), *The Mental Representation of Gramatical Relations*, Cambridge, MA: MIT Press.

Frazier, L. & Fodor, J. D. (1978). The Sausage Machine: A new two-stage parsing model, *Cognition* 6, 291-325.

Frazier, L. & Rayner, K. (1982). Making and Correcting Errors during Sentence Comprehension: Eye Movementsin the Analysis of Structurally Ambiguous Sentences, *Cognitive Psychology* 14, 178-210.

Hindle, D. (1989). Acquiring disambiguation rules from text. In *Proceedings of 27th Annual Meeting of the Association for Computational Linguistics.*

Hobbs, J.R. & Bear, J. (1990). Two Principles of Parsing Preferences. In *Proceedings of the Thirteenth International Conference on Computational Linguistics.* Helsinki.

Huyck, C.R. & Lytinen, S. L. (1993). Efficient Heuristic Natural Language Parsing. In Proceedings of the National Conference on Artificial Intelligence (pp. 386-391).

Kimball, J. P. (1973). Seven principles of surface structure parsing in natural language. *Cognition*, 2, 15-47.

Mahesh, K. & Eiselt, K. P. (1994). Uniform Representations for Syntax-Semantics Arbitration. In *Proceedings of the Sixteenth Annual Meeting of the Cognitive Science Society* (pp. 589-594). Atlanta, GA: Lawrence Erlbaum Associates.

Milne, R. (1982). Predicting Garden Path Sentences. *Cognitive Science* 6, 349-373.

Milne, R. (1986). Resolving ambiguities in a deterministic parser. *Computational Linguistics* 12, 1-12.

Shieber, S. M. (1983). Sentence disambiguation by a shift-reduce parsing technique. In *Proceedings of 21st Annual Meeting of the Association for Computational Linguistics* (pp. 113-118). Cambridge, MA.

Stevenson, S. (1994). A Unified Model of Preference and Recovery Mechanisms in Human Parsing. In *Proceedings of the Sixteenth Annual Meeting of the Cognitive Science Society* (pp. 824-829). Atlanta, GA: Lawrence Erlbaum Associates.

Tomita, M. (1987). An Efficient Augmented-Context-Free Parsing Algorithm. *Computational Linguistics* 13, 31-46.

Wiren, M. (1990). Incremental Parsing and Reason Maintenance. In *Proceedings of the Thirteenth International Conference on Computational Linguistics* (pp. 287-292). Helsinki.

Zernik, U. & Brown, A. (1988). Default Reasoning in Natural Language Processing. In *Proceedings of the Twelfth International Conference on Computational Linguistics* (pp. 801-805). Budapest.