UNGRAMMATICALITY AND EXTRA-GRAMMATICALITY IN NATURAL LANGUAGE UNDERSTANDING SYSTEMS

By

Stan C. Kwasny **
The Ohio State University
Columbus, Ohio

Norman K. Sondheimer
Sperry Univac
Blue Bell, Pennsylvania

I. Introduction

Among the components included in Natural Language Understanding (NLU) systems is a grammar which specifies much of the linguistic structure of the utterances that can be expected. However, it is certain that inputs that are ill-formed with respect to the grammar will be received, both because people regularly form ungrammatical utterances and because there are a variety of forms that cannot be readily included in current grammatical models and are hence "extra-grammatical". These might be rejected, but as Wilks stresses, "...understanding requires, at the very least, ... some attempt to interpret, rather than merely reject, what seem to be ill-formed utterances." [WIL76]

This paper investigates several language phenomena commonly considered ungrammatical or extra-grammatical and proposes techniques directed at integrating them as much as possible into the conventional grammatical processing performed by NLU systems through Augmented Transition Network (ATN) grammars. For each NLU system, a "normative" grammar is assumed which specifies the structure of well-formed inputs. Rules that are both manually added to the original grammar or automatically constructed during parsing analyze the ill-formed input. The ill-formedness is shown at the completion of a parse by deviance from fully grammatical structures. We have been able to do this processing while preserving the structural characteristics of the original grammar and its inherent efficiency.

Some of the phenomena discussed have been considered previously in particular NLU systems, see for example the ellipsis handling in LIFER [HEN77]. Some techniques similar to ours have been used for parsing, see for example the conjunction mechanism in LUNAR [WOO73]. On the linguistic side, Chomsky [CHO64] and Katz [KAT64], among others have considered the treatment of ungrammaticality in Transformational Grammar theories. The study closest to ours is that of Weischedel and Black [WEI79]. The present study is distinguished by the range of phenomena considered, its structural and efficiency goals, and the inclusion of the techniques proposed within one implementation.

This paper looks at these problems, proposes mechanisms aimed at solving the problems, and describes how these mechanisms are used. At the end, some extensions are suggested. Unless otherwise noted, all ideas have been tested through implementation. A more detailed and extended discussion of all points may be found in Kwasny [KWA79].

II. Language Phenomena

Success in handling ungrammatical and extra-grammatical input depends on two factors. The first is the identification of types of ill-formedness and the patterns they follow. The second is the relating of ill-formed input to the parsing path of a grammatical input the user intends. This section introduces the types of ill-formedness we have studied, and discusses their relationship to grammatical structures in terms of ATN grammars.

** Current Address:
Computer Science Department
Indiana University
Bloomington, Indiana

II.1 Co-Occurrence Violations

Our first class of errors can be connected to co-occurrence restrictions within a sentence. There are many occassions in a sentence where two parts or more must agree (* indicates an ill-formed or ungrammatical sentence):

*Draw a circles.

*I will stay from now under midnight.

The errors in the above involve coordination between the underlined words. The first example illustrates simple agreement problems. The second involves a complicated relation between at least the three underlined terms.

Such phenomena do occur naturally. For example, Shore [SHO77] analyzes fifty-six freshman English papers written by Black college students and reveals patterns of nonstandard usage ranging from uninflected plurals, possessives, and third person singulars to overinflection (use of inappropriate endings.)

For co-occurrence violations, the blocks that keep inputs from being parsed as the user intended arise from a failure of a test on an arc or the failure to satisfy an arc type restriction, e.g., failure of a word to be in the correct category. The essential block in the first example would likely occur on an agreement test on an arc accepting a noun. The essential blockage in the second example is likely to come from failure of the arc testing the final preposition.

II.2 Ellipsis and Extraneous Terms

In handling ellipsis, the most relevant distinction to make is between contextual and telegraphic ellipsis.

Contextual ellipsis occurs when a form only makes proper sense in the context of other sentences. For example, the form

*President Carter has.

seems ungrammatical without the preceding question form

Who has a daughter named Amy?

President Carter has.

Telegraphic ellipsis, on the other hand, occurs when a form only makes proper sense in a particular situation. For example, the forms

3 chairs no waiting (sign in barber shop)

Yanks split (headline in sports section)

Profit margins for each product
(query submitted to a NLU system)

are cases of telegraphic ellipsis with the situation noted in parentheses. The final example is from an experimental study of NLU for management information which indicated that such forms must be considered [MAL75].

Another type of ungrammaticality related to ellipsis occurs when the user puts unnecessary words or phrases in an utterance. The reason for an extra word may be a change of intention in the middle of an utterance, an oversight, or simply for emphasis. For example,

*Draw a line with from here to there.

*List prices of single unit prices for 72 and 73.

The second example comes from Malhotra [MAL75].

The best way to see the errors in terms of the ATN is to think of the user as trying to complete a path through the grammar, but having produced an input that has too many or too few forms necessary to traverse all arcs.

II.3 Conjunction

Conjunction is an extremely common phenomenon, but it is seldom directly treated in a grammar. We have considered several types of conjunction.

Simple forms of conjunction occur most frequently, as in

John loves Mary and hates Sue.

Gapping occurs when internal segments of the second conjunct are missing, as in

John loves Mary and Mary John.

The list form of conjunction occurs when more than two elements are joined in a single phrase, as in

John loves Mary, Sue, Nancy, and Bill.

Correlative conjunction occurs in sentences to coordinate the joining of constituents, as in

John both loves and hates Sue.

The reason conjuncts are generally left out of grammars is that they can appear in so many places that inclusion would dramatically increase the size of the grammar. The same argument applies to the ungrammatical phenomena. Since they allow so much variation compared to grammatical forms, including them with existing techniques would dramatically increase the size of a grammar. Further there is a real distinction in terms of completeness and clarity of intent between grammatical and ungrammatical forms. Hence we feel justified in suggesting special techniques for their treatment.

III. Proposed Mechanisms and How They Apply

The following presentation of our techniques assumes an understanding of the ATN model. The techniques are applied to the language phenomena discussed in the previous section.

III.1 Relaxation Techniques

The first two methods described are relaxation methods which allow the successful traversal of ATN arcs that might not otherwise be traversed. During parsing, whenever an arc cannot be taken, a check is made to see if some form of relaxation can apply. If it can, then a backtrack point is created which includes the relaxed version of the arc. These alternatives are not considered until after all possible grammatical paths have been attempted thereby insuring that grammatical inputs are still handled correctly. Relaxation of previously relaxed arcs is also possible. Two methods of relaxation have been investigated.

Our first method involves relaxing a test on an arc, similar to the method used by Weischedel in [WEI79]. Test relaxation occurs when the test portion of an arc contains a relaxable predicate and the test fails. Two methods of test relaxation have been identified and implemented based on predicate type. Predicates can be designated by the grammar writer as either absolutely violable in which case the opposite value of the predicate (determined by the LISP function NOT applied to the predicate) is substituted for the predicate during relaxation or conditionally violable in which case a substitute predicate is provided. For example, consider the following to be a test that fails:

```
(AND
     (INFLECTING  V)
     (INTRANS  V))
```

If the predicate INFLECTING was declared absolutely violable and its use in this test returned the value NIL, then the negation of (INFLECTING V) would replace it in the test creating a new arc with the test:

```
(AND
     T
     (INTRANS  V))
```

If INTRANS were conditionally violable with the substitute predicate TRANS, then the following test would appear on the new arc:

```
(AND
     (INFLECTING  V)
     (TRANS  V))
```

Whenever more than one test in a failing arc is violable, all possible single relaxations are attempted independently. Absolutely violable predicates can be permitted in cases where the test describes some superficial consistency checking or where the test's failure or success doesn't have a direct affect on meaning, while conditionally violable predicates apply to predicates which must be relaxed cautiously or else loss of meaning may result.

Chomsky discusses the notion of organizing word categories hierarchically in developing his ideas on degrees of grammaticalness. We have applied and extended these ideas in our second method of relaxation called category relaxation. In this method, the grammar writer produces, along with the grammar, a hierarchy describing the relationship among words, categories, and phrase types which is utilized by the relaxation mechanism to construct relaxed versions of arcs that have failed. When an arc fails because of an arc type failure (i.e., because a particular word, category, or phrase was not found) a new arc (or arcs) may be created according to the description of the word, category, or phrase in the hierarchy. Typically, PUSH arcs will relax to PUSH arcs, CAT arcs to CAT or PUSH arcs, and WRD or MEM arcs to CAT arcs. Consider, for example, the syntactic category hierarchy for pronouns shown in Figure 1. For this example, the category relaxation

mechanism would allow the relaxation of PERSONAL pronouns to include the category PRONOUN. The arc produced from category relaxation of PERSONAL pronouns also includes the subcategories REFLEXIVE and DEMONSTRATIVE in order to expand the scope of terms during relaxation. As with test relaxation, successive relaxations could occur.

For both methods of relaxation, "deviance notes" are generated which describe the nature of the relaxation in each case. Where multiple types or multiple levels of relaxation occur, a note is generated for each of these. The entire list of deviance notes accompanies the final structure produced by the parser. In this way, the final structure is marked as deviant and the nature of the deviance is available for use by other components of the understanding system.

In our implementation, test relaxation has been fully implemented, while category relaxation has been implemented for all cases except those involving PUSH arcs. Such an implementation is anticipated, but requires a modification to our backtracking algorithm.

III.2 Co-Occurrence and Relaxation

The solution being proposed to handled forms that are deviant because of co-occurrence violations centers around the use of relaxation methods. Where simple tests exist within a grammar to filter out unacceptable forms of the type noted above, these tests may be relaxed to allow the acceptance of these forms. This doesn't eliminate the need for such tests since these tests help in disambiguation and provide a means by which sentences are marked as having violated certain rules.

For co-occurrence violations, the point in the grammar where parsing becomes blocked is often exactly where the test or category violation occurs. An arc at that point is being attempted and fails due to a failure of the co-occurrence test or categorization requirements. Relaxation is then applied and an alternative generated which may be explored at a later point via backtracking. For example, the sentence:

*John love Mary

shows a disagreement between the subject (John) and the verb (love). Most probably this would show up during parsing when an arc is attempted which is expecting the verb of the sentence. The test would fail and the traversal would not be allowed. At that point, an ungrammatical alternative is created for later backtracking to consider.

III.3 Patterns and the Pattern Arc

In this section, relaxation techniques, as applied to the grammar itself, are introduced through the use of patterns and pattern-matching algorithms. Other systems have used patterns for parsing. We have devised a powerful method of integrating, within the ATN formalism, patterns which are flexible and useful.

In our current formulation, which we have implemented and are now testing, a pattern is a linear sequence of ATN arcs which is matched against the input string. A pattern arc (PAT) has been added to the ATN formalism whose form is similar to that of other arcs:

(PAT <pat spec> <test> <act>* <term>)

The pattern specification (<pat spec>) is defined as:

<pat spec> ::= (<patt> <mode>*)

<patt> ::= (<p arc>*)
<pat name>

<mode> ::= UNANCHOR
OPTIONAL
SKIP

<p arc> ::= <arc>
> <arc>

<pat name> ::= user-assigned pattern name
>

The pattern (<patt>) is either the name of a pattern, a ">", or a list of ATN arcs, each of which may be preceded by the symbol ">", while the pattern mode (<mode>) can be any of the keywords, UNANCHOR, OPTIONAL, or SKIP. These are discussed below. To refer to patterns by name, a dictionary of patterns is supported. A dictionary of arcs is also supported, allowing the referencing of arcs by name as well. Further, named arcs are defined as macros, allowing the dictionary and the grammar to be substantially reduced in size.

THE PATTERN MATCHER

Pattern matching proceeds by matching each arc in the pattern against the input string, but is affected by the chosen "mode" of matching. Since the individual component arcs are, in a sense, complex patterns, the ATN interpreter can be considered part of the matching algorithm as well. In arcs within patterns, explicit transfer to a new state is ignored and the next arc attempted on success is the one following in the pattern. An arc in a pattern prefaced by ">" can be considered optional, if the OPTIONAL mode has been selected to activate this feature. When this is done, the matching algorithm still attempts to match optional arcs, but may ignore them. A pattern unanchoring capability is activated by specifying the mode UNANCHOR. In this mode, patterns are permitted to skip words prior to matching. Finally, selection of the SKIP mode results in words being ignored between matches of the arcs within a pattern. This is a generalization of the UNANCHOR mode.

Pattern matching again results in deviance notes. For patterns, they contain information necessary to determine how matching succeeded.

SOURCE OF PATTERNS

An automatic pattern generation mechanism has been implemented using the trace of the current execution path to produce a pattern. This is invoked by using a ">" as the pattern name. Patterns produced in this fashion contain only those arcs traversed at the current level of recursion in the network, although we are planning to implement a generalization of this in which PUSH arcs can be automatically replaced by their subnetwork paths. Each arc in an automatic pattern is marked as optional. Patterns can also be constructed dynamically in precisely the same way grammatical structures are built using BUILDQ. The vehicle by which this is accomplished is discussed next.

AUTOMATIC PRODUCTION OF ARCS

Pattern arcs enter the grammar in two ways. They are manually written into the grammar in those cases where the ungrammaticalities are common and they are added to the grammar automatically in those cases where the ungrammaticality is dependent on context. Pattern arcs produced dynamically enter the grammar through one of two devices. They may be constructed as needed by

special macro arcs or they may be constructed for future use through an expectation mechanism.

As the expectation-based parsing efforts clearly show, syntactic elements especially words contain important clues on processing. Indeed, we also have found it useful to make the ATN mechanism more "active" by allowing it to produce new arcs based on such clues. To achieve this, the CAT, MEM, TST, and WRD arcs have been generalized and four new "macro" arcs, known as CAT*, MEM*, TST*, and WRD*, have been added to the ATN formalism. These are similar in every way to their counterparts, except that as a final action, instead of indicating the state to which the traversal leads, a new arc is constructed dynamically and immediately executed. The difference in the form that the new arc takes is seen in the following pair where <creat act> is used to define the dynamic arc:

```
(CAT  <cat> <test> <act>* <term     >)
(CAT* <cat> <test> <act>* <creat act>)
```

Arcs computed by macro arcs can be of any type permitted by the ATN, but one of the most useful arcs to compute in this manner is the PAT arc discussed above.

EXPECTATIONS

The macro arc forces immediate execution of an arc. Arcs may also be computed and temporarily added to the grammar for later execution through an "expectation" mechanism. Expectations are performed as actions within arcs (analogous to the HOLD action for parsing structures) or as actions elsewhere in the NLU system (e.g., during generation when particular types of responses can be foreseen). Two forms are allowed:

```
(EXPECT <creat act> <state>)
(EXPECT <creat act>        )
```

In the first case, the arc created is bound to a state as specified. When later processing leads to that state, the expected arc will be attempted as one alternative at that state. In the second case, where no state is specified, the effect is to attempt the arc at every state visited during the parse.

The range of an expectation produced during parsing is ordinarily limited to a single sentence, with the arc disappearing after it has been used; however, the start state, S*, is reserved for expectations intended to be active at the beginning of the next sentence. These will disappear in turn at the end of processing for that sentence.

III.4 Patterns, Ellipsis, and Extraneous Forms

The Pattern arc is proposed as the primary mechanism for handling ellipsis and extraneous forms. A Pattern arc can be seen as capturing a single path through a network. The matcher gives some freedom in how that path relates to a string. We propose that the appropriate parsing path through a network relates to an elliptical sentence or one with extra words in the same way. With contextual ellipsis, the relationship will be in having some of the arcs on the correct path not satisfied. In Pattern arcs, these will be represented by arcs marked as optional. With contextual ellipsis, dialogue context will provide the defaults for the missing components. With Pattern arcs, the deviance notes will show what was left out and the other components in the NLU system will be responsible for supplying the values.

The source of patterns for contextual ellipsis is important. In Lifer [HEN77], the previous user input can be seen as a pattern for elliptical processing of the current input. The automatic pattern generator developed here, along with the expectation mechanism, will capture this level of processing. But with the ability to construct arbitrary patterns and to add them to the grammar from other components of the NLU system, our approach can acccomplish much more. For example, a question generation routine could add an expectation of a yes/no answer in front of a transformed rephrasing of a question, as in

Did Amy kiss anyone?

Yes, Jimmy was kissed.

Patterns for telegraphic ellipsis will have to be added to the grammar manually. Generally, patterns of usage must be identified, say in a study like that of Malhotra, so that appropriate patterns can be constructed. Patterns for extraneous forms will also be added in advance. These will either use the unachor option in order to skip false starts, or dynamically produced patterns to catch repetitions for emphasis. In general, only a limited number of these patterns should be required. The value of the pattern mechanism here, especially in the case of telegraphic ellipsis, will be in connecting the ungrammatical to grammatical forms.

III.5 Conjunction and Macro Arcs

Pattern arcs are also proposed as the primary mechanism for handling conjunction. The rationale for this is the often noted connection between conjunction and ellipsis, see for example Halliday and Hasan [HAL76]. This is clear with gapping, as in the following where the parentheses show the missing component

John loves Mary and Mary (loves) John.

But it also can be seen with other forms, as in

John loves Mary and (John) hates Sue.

John loves Mary, (John loves) Sue, (John loves) Nancy, and (John loves) Bill.

Whenever a conjunction is seen, a pattern is developed from the already identified elements and matched against the remaining segments of input. The heuristics for deciding from which level to produce the pattern force the most general interpretation in order to encourage an elliptical reading.

All of the forms of conjunction described above are treated through a globally defined set of "conjunction arcs" (Some restricted cases, such as "and" following "between", have the conjunction built into the grammar). In general, this set will be made up of macro arcs which compute Pattern arcs. The automatic pattern mechanism is heavily used. With simple conjunctions, the rightmost elements in the patterns are matched. Internal elements in patterns are skipped with gapping. The list form of conjunction can also be handled through the careful construction of dynamic patterns which are then expected at a later point. Correlatives are treated similarly, with expectations based on the dynamic building of patterns.

There are a number of details in our proposal which will not be presented. There are also visible limits. It is instructive to compare the proposal to the SYSCONJ facility of Woods [WOO73]. It treats conjunction as

showing alternative ways of continuing a sentence. This allows for sentences such as

> He drove his car through and broke a plate glass window.

which at best we will accept with a misleading deviance note. However, it can not handle the obvious elliptical cases, such gapping, or the tightly constrained cases, such as correlatives. We expect to continue investigating the pattern approach.

III.6 Interaction of Techniques

As grammatical processing proceeds, ungrammatical possibilities are continually being suggested from the various mechanisms we have implemented. To coordinate all of these activities, the backtracking mechanism has been improved to keep track of these alternatives. All paths in the original grammar are attempted first. Only when these all fail are the conjunction alternatives and the manually added and dynamically produced ungrammatical alternatives tried. All of the alternatives of these sorts connected with a single state can be thought of as a single possibility. A selection mechanism is used to determine which backtrack point among the many potential alternatives is worth exploring next. Currently, we use a method also used by Weischedel and Black [WEI79] of selecting the alternative with the longest path length.

IV. Conclusion and Open Questions

These results are significant, we believe, because they extend the state of the art in several ways. Most obvious are the following:

> The use of the category hierarchy to handle arc type failures;
>
> The use of the pattern mechanism to allow for contextual ellipsis and gapping;
>
> More generally, the use of patterns to allow for many sorts of ellipsis and conjunctions; and
>
> Finally, the orchestration of all of the techniques in one coherent system, where because all grammatical alternatives are tried first and no modifications are made to the original grammar, its inherent efficiency and structure are preserved.

IV.1 Open Problems

Various questions for further research have arisen during the course of this work. The most important of these are discussed here.

Better control must be exercised over the selection of viable alternatives when ungrammatical possibilities are being attempted. The longest-path heuristic is somewhat weak. The process that decides this would need to take into consideration, among other things, whether to allow relaxation of a criteria applied to the subject or to the verb in a case where the subject and verb do not agree. The current path length heuristic would always relax the verb which is clearly not always correct.

No consideration has been given to the possible connection of one error with another. In some cases, one error can lead to or affect another.

Several other types of ill-formedness have not been considered in this study, for example, idioms, metaphors, incorrect word order, run together sentences, incorrect punctuation, misspelling, and presuppositional failure. Either little is known about these processes or they have been studied elsewhere independently. In either case, work remains to be done.

V. Acknowledgments

We wish to acknowledge the comments of Ralph Weischedel and Marc Fogel on previous drafts of this paper. Although we would like to blame them, any shortcomings are clearly our own fault.

VI. Bibliography

[CHO64] Chomsky, N., "Degrees of Grammaticalness," in [FOD64], 384-389.

[FOD64] Fodor, J. A. and J. J. Katz, The Structure of Language: Readings in the Philosophy of Language, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.

[HAL76] Halliday, M.A.K. and R. Hasan, Cohesion in English, Longman, London, 1976.

[HEN77] Hendrix, G. G., "The LIFER Manual," Technical Note 138, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California, February, 1977.

[KAT64] Katz, J. J., "Semi-Sentences," in [FOD64], 400-416.

[KWA79] Kwasny, S., "Treatment of Ungrammatical and Extragrammatical Phenomena in Natural Language Understanding Systems," PhD dissertation (forthcoming), Ohio State University, 1979.

[MAL75] Malhotra, A., "Design Criteria for a Knowledge-Based English Language System for Management: An Experimental Analysis," MAC TR-146, M.I.T., Cambridge, Ma, February, 1975.

[SHO77] Shores, D. L., "Black English and Black Attitudes," in Papers in Language Variation, D. L. Shores and C. P. Hines (Ed.), The University of Alabama Press, University, Alabama, 1977.

[WEI79] Weischedel, R. M., and J. Black, "Responding to Potentially Unparseable Sentences," manuscript, Department of Computer and Information Sciences, University of Delaware, Newark, Delaware, 1979.

[WIL76] Wilks, Y., "Natural Language Understanding Systems Within the A.I. Paradigm: A Survey," American Journal of Computational Linguistics, microfiche #40, 1, 1976.

[WOO73] Woods, W. A. "An Experimental Parsing System for Transition Network Grammars," in Natural Language Processing, R. Rustin (Ed.), Algorithmics Press, 1973.

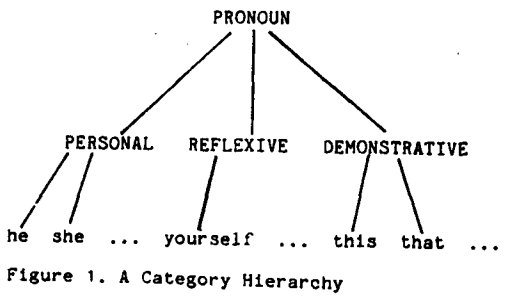

Figure 1. A Category Hierarchy