

# An Improved Heuristic for Ellipsis Processing\*

Ralph M. Weischedel  
Department of Computer & Information Sciences  
University of Delaware  
Newark, Delaware 19711

and  
Norman K. Sondheimer  
Software Research  
Sperry Univac MS 2G3  
Blue Bell, Pennsylvania 19424

## 1. Introduction

Robust response to ellipsis (fragmentary sentences) is essential to acceptable natural language interfaces. For instance, an experiment with the REL English query system showed 10% elliptical input (Thompson, 1980).

In Quirk, et al. (1972), three types of contextual ellipsis have been identified:

1. repetition, if the utterance is a fragment of the previous sentence.
2. replacement, if the input replaces a structure in the previous sentence.
3. expansion, if the input adds a new type of structure to those used in the previous sentence.

Instances of the three types appear in the following example.

Were you angry?

- a) I was. (repetition with change in person)
- b) Furious. (replacement)
- c) Probably. (expansion)
- d) For a time. (expansion)
- e) Very. (expansion)
- f) I did not want to be. (expansion)
- g) Yesterday I was. (expansion & repetition)

In addition to appearing as answers following questions, any of the three types can appear in questions following statements, statements following statements, or in the utterances of a single speaker.

This paper presents a method of automatically interpreting ellipsis based on dialogue context. Our method expands on previous work by allowing for expansion ellipsis and by allowing for all combinations of statement following question, question following statement, question following question, etc.

## 2. Related Work

Several natural language systems (e.g., Bobrow et al., 1977; Hendrix et al., 1978; Kwasny and Sondheimer, 1979) include heuristics for replacement and repetition ellipsis, but not expansion ellipsis. One general strategy has been to substitute fragments into the analysis of the previous input, e.g., substituting parse trees of the elliptical input into the parse trees of the previous input in LIFER (Hendrix, et al., 1978). This only applies to inputs of the same type, e.g., repeated questions.

Allen (1979) deals with some examples of expansion ellipsis, by fitting a parsed elliptical input into a model of the speaker's plan. This is similar to other methods that interpret fragments by placing them into prepared fields in frames or case slots (Schank et al., 1980; Hayes and Mouradian, 1980; Waltz, 1978). This approach seems most applicable to limited-domain systems.

## 3. The Heuristic

There are three aspects to our solution: a mechanism for repetition and replacement ellipsis, an extension for inputs of different types, such as fragmentary answers to questions, and an extension for expansion ellipsis.

### 3.1 Repetition and Replacement

As noted above, repetition and replacement ellipsis can be viewed as substitution in the previous form. We have implemented this notion in an augmented transition network (ATN) grammar interpreter with the assumption that the "previous form" is the complete ATN path that parsed the previous input and that the lexical items consumed along that path are associated with the arcs that consumed them. In ellipsis mode, the ATN interpreter executes the path using the elliptical input in the following way:

-----  
\*This material is based upon work partially supported by the National Science Foundation under Grant No. IST-8009673.

1. Words from the elliptical input, i.e., the current input, may be consumed along the path at any point.
2. Any arc requiring a word not found in the current input may be traversed using the lexical item associated with the arc from the previous input.
3. However, once the path consumes the first word from the elliptical input, all words from the elliptical input must be consumed before an arc can use a word from the previous input.
4. Traversing a PUSH arc may be accomplished either by following the subpath of the previous input or by finding any constituent of the required type in the current input. The entire ATN can be used in these cases.

Suppose that the path for "Were you angry?" is given by Table 1. Square brackets are used to indicate subpaths resulting from PUSHes. "..." indicates tests and actions which are irrelevant to the current discussion.

| <u>State</u> | <u>Arc</u>               | <u>Old Lexical Item</u> |
|--------------|--------------------------|-------------------------|
| S            | (CAT COPULA ... (TO Sx)) | "were"                  |
| Sx           | (PUSH NP ... (TO Sy))    |                         |
| [NP          | (CAT PRO ... (TO NPa))   | "you"                   |
| NPa          | (POP ...)                |                         |
| Sy           | (CAT ADJ ... (TO Sz))    | "angry"                 |
| Sz           | (POP ...)                |                         |

Table 1  
An ATN Path for "Were you Angry?"

An elliptical input of "Was he?" following "Were you angry?" could be understood by traversing all of the arcs as in Table 1. Following point 1 above, "was" and "he" would be substituted for "were" and "you". Following point 3, in traversing the arc (CAT ADJ ... (TO Sz)) the lexical item "angry" from the previous input would be used. Item 4 is illustrated by an elliptical input of "Was the old man?"; this is understood by traversing the arcs at the S level of Table 1, but using the appropriate path in the NP network to parse "the old man".

### 3.2 Transformations of the Previous Form

While the approach illustrated in Section 3.1 is useful in a data base query environment where elliptical input typically is a modification of the previous query, it does not account for elliptical statements following questions, elliptical

questions following statements, etc. Our approach to the problem is to write a set of transformations which map the parse path of a question (e.g., Table 1) into an expected parse path for a declarative response, and the parse path for a declarative into a path for an expected question, etc.

The left-hand side of a transformation is a pattern which is matched against the ATN path of the previous utterance. Pattern elements include literals referring to arcs, variables which match a single arc or embedded path, variables which match zero or more arcs, and sets of alternatives. It is straightforward to construct a discrimination net corresponding to all left-hand sides for efficiently finding what patterns match the ATN path of the previous sentence. The right-hand side of a transformation is a pattern which constructs an expected path. The form of the pattern on the right-hand side is a list of references to states, arcs, and lexical entries. Such references can be made through items matched on the left-hand side or by explicit construction of literal path elements.

Our technique is to restrict the mapping such that any expected parse path is generated by applying only one transformation and applying it only once. A special feature of our transformational system is the automatic allowance for dialogue deixis. An expected parse path for the answer to "Were you angry?" is given in Table 2. Note in Table 2, "you" has become "I" and "were" has become "was".

| <u>State</u> | <u>Arc</u>               | <u>Old Lexical Item</u> |
|--------------|--------------------------|-------------------------|
| S            | (PUSH NP ... (TO Sa))    |                         |
| [NP          | (CAT PRO ... (TO NPa))   | "I"                     |
| NPa          | (POP ...)                |                         |
| Sa           | (CAT COPULA ... (TO Sy)) | "was"                   |
| Sy           | (CAT ADJ ... (TO Sz))    | "angry"                 |
| Sz           | (POP ...)                |                         |

Table 2  
Declarative for the expected answer for "Were you angry?".

Using this path, the ellipsis interpreter described in Section 3.1 would understand the ellipses in "a)" and "b)", below, in the same way as "a')'" and "b')'".

- a) I was.
- a') I was angry.
- b) My spouse was.
- b') My spouse was angry.

### 3.3 Expansions

A large class of expansions are simple adjuncts, such as examples c, d, e, and g in section 1. We have handled this by building our ellipsis interpreter to allow departing from the base path at designated states to consume an adjunct from the input string. We mark states in the grammar where adjuncts can occur. For each such state, we list a set of linear (though possibly cyclic) paths, called "expansion paths". Our interpreter as implemented allows departures from the base path at any state so marked in the grammar; it follows expansion paths by consuming words from the input string, and must return to a state on the base form. Each of the examples in c, d, e, and g of section 1 can be handled by expansion paths only one arc long. They are given in Table 3.

#### Initial

| <u>State</u> | <u>Expansion Path</u>   |
|--------------|---|
| S            | (PUSH ADVERB ... (TO S))<br>Probably (I was angry).   |
| S            | (PUSH PP ... (TO S))<br>For a time (I was angry).   |
| S            | (PUSH NP<br>(* this includes a test<br>that the NP is one<br>of time or place)<br>... (TO S))<br>Yesterday (I was angry). |
| Sy           | (PUSH INTENSIFIER-ADVERB<br>... (TO Sy))<br>(I was) very (angry).   |

Table 3  
Example Expansion Paths

Since this is an extension to the ellipsis interpreter, combinations of repetition, replacement, and expansion can all be handled by the one mechanism. For instance, in response to "Were you angry?", "Yesterday you were (angry)" would be treated using the expansion and replacement mechanisms.

### 4. Special Cases and Limitations

The ideal model of contextual ellipsis would correctly predict what are appropriate elliptical forms in context, what their interpretation is, and what forms are not meaningful in context. We believe this requires structural restrictions, semantic constraints, and a model of the goals of the speaker. Our heuristic does not meet these criteria in a number of cases.

Only two classes of structural constraints are captured. One relates the ellipsis to the previous form as a combination of repetition, replacement, and expansion. The other constraint is that the input must be consumed as a contiguous string. This constraint is violated, for instance, in "I was (angry) yesterday" as a response to "Were you angry?". Nevertheless, the constraint is computationally useful, since allowing arbitrary gaps in consuming the elliptical input produces a very large space of correct interpretations. A ludicrous example is the following question and elliptical response:

Has the boss given our mutual friend a raise?  
A fat raise.

Allowing arbitrary gaps between the substrings of the ellipsis allows an interpretation such as "A (boss has given our) fat (friend a) raise."

While it may be possible to view all contextual ellipsis as combinations of the operations repetition, replacement, and expansion applied to something, our model makes the strong assumption that these operations may be viewed as applying to an ATN path rather straightforwardly related to the previous utterance. Not all expansions can be viewed that way, as example f in Section 1 illustrates. Also, answers of "No" require special processing; that response in answer to "Were you angry" should not be interpreted as "No, I was angry." One should be able to account for such examples within the heuristic described in this paper, perhaps by allowing the transformation system described in section 3.2 to be completely general rather than strongly restricted to one and only one transformation application. However, we propose handling such cases by special purpose rules we are developing. These rules for the special cases, plus the mechanism described in section 3 together will be formally equivalent in predictive power to a grammar for elliptical forms.

Though the heuristic is independent of the individual grammar, designating expansion paths and transformations obviously is not. The grammar may make this an easy or difficult task. For instance in the grammar we are using, a subnetwork that collects all tense, aspect, and modality elements would simplify some of the transformations and expansion paths.

Naturally, semantics must play an important part in ellipsis processing. Consider the utterance pair below:

Did the boss have a martini at lunch?  
Some wine.

Though syntactically this could be interpreted either as "Some wine (did have a martini at lunch)", "(The boss did have) some wine (at lunch)", or "(The boss did have a martini at) some wine". Semantics should prefer the second reading. We are testing our heuristic using the RUS grammar (Bobrow, 1978) which has frequent calls from the grammar requesting that the semantic component decide whether to build a semantic interpretation for the partial parse found or to veto that partial parse. This should aid performance.

## 5. Summary and Conclusion

There are three aspects to our solution: a mechanism for repetition and replacement ellipsis, an extension for inputs of different types, such as fragmentary answers to questions, and an extension for expansion ellipsis.

Our heuristic deals with the three types of expansion ellipsis as follows: Repetition ellipsis is processed by repeating specific parts of a transformed previous path using the same phrases as in the transformed form ("I was angry"). Replacement ellipsis is processed by substituting the elliptical input for contiguous constituents on a transformed previous path. Expansion ellipsis may be processed by taking specially marked paths that detour from a given state in that path. Combinations of the three types of ellipsis are represented by combinations of the three variations in a transformed previous path.

There are two contributions of the work. First, our method allows for expansion ellipsis. Second, it accounts for combinations of previous sentence form and ellided form, e.g., statement following question, question following statement, question following question. Furthermore, the method works without any constraints on the ATN grammar. The heuristics carry over to formalisms similar to the ATN, such as context-free grammars and augmented phrase structure grammars.

Our study of ellipsis is part of a much broader framework we are developing for processing syntactically and/or semantically ill-formed input; see Weischedel and Sondheimer (1981).

## References

- Allen, James F., "A Plan-Based Approach to Speech Act Recognition," Ph.D. Thesis, Dept. of Computer Science, University of Toronto, Toronto, Canada, 1979.
- Bobrow, D., R. Kaplan, M. Kay, D. Norman, H. Thompson and T. Winograd, "GUS, A Frame-driven Dialog System", Artificial Intelligence, 8, (1977), 155-173.
- Bobrow, R., "The RUS System", in Research in Natural Language Understanding, by B. Webber and R. Bobrow, BBN Report No. 3878, Bolt Beranek and Newman, Inc., Cambridge, MA, 1978.
- Hayes, P. and G. Mouradian, "Flexible Parsing", in Proc. of the 18th Annual Meeting of the Assoc. for Comp. Ling., Philadelphia, June, 1980, 97-103.
- Hendrix, G., E. Sacerdoti, D. Sagalowicz and J. Slocum, "Developing a Natural Language Interface to Complex Data", ACM Trans. on Database Sys., 3, 2, (1978), 105-147.
- Kwasny, S. and N. Sondheimer, "Ungrammaticality and Extragrammaticality in Natural Language Understanding Systems", in Proc. of the 17th Annual Meeting of the Assoc. for Comp. Ling., San Diego, August, 1979, 19-23.
- Quirk, R., S. Greenbaum, G. Leech and J. Svartvik, A Grammar of Contemporary English, Seminar Press, New York, 1972.
- Schank, R., M. Lebowitz and L. Birnbaum, "An Integrated Understander", American Journal of Comp. Ling., 6, 1, (1980), 13-30.
- Thompson, B. H., "Linguistic Analysis of Natural Language Communication with Computers", Proceedings of the Eighth International Conference on Computational Linguistics, Tokyo, October, 1980, 190-201.
- Waltz, D., "An English Language Question Answering System for a Large Relational Database", Comm. ACM, 21, 7, (1978), 526-539.
- Weischedel, Ralph M. and Norman K. Sondheimer, "A Framework for Processing Ill-formed Input", Technical Report, Dept. of Computer & Information Sciences, University of Delaware, Newark, DE, 1981.

## Acknowledgement

Much credit is due to Amir Razi for his programming assistance.