ARPA ORDER NO. 2223

ISI/RR-78-72
October 1978

AD A061 681

LEVEL II

James A. Levin
Neil M. Goldman

Process Models of Reference in Context

DDC FILE COPY

INFORMATION SCIENCES INSTITUTE
UNIVERSITY OF SOUTHERN CALIFORNIA
4676 Admiralty Way/Marina del Rey/California 90291
(213) 822-1511

78 11 24

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ISI/RR-78-72	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Process Models of Reference in Context.		5. TYPE OF REPORT & PERIOD COVERED Research rept.
		6. PERFORMING ORG. REPORT NUMBER ARPA Order-2223
7. AUTHOR(s) James A. Levin Neil M. Goldman		8. CONTRACT OR GRANT NUMBER(s) NSF-MCS76-07332
9. PERFORMING ORGANIZATION NAME AND ADDRESS Information Sciences Institute 4676 Admiralty Way Marina del Rey, CA 90291		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Mathematical and Computer Sciences Division National Science Foundation 1800 G St. NW Washington, D. C. 20550		12. REPORT DATE October 1978
		13. NUMBER OF PAGES 28
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 29p.		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document approved for public release and sale; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

James Levin is now located at the Communications Program, University of California at San Diego, La Jolla, CA 92093.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

anaphoric reference, artificial intelligence, context models, natural language generation, natural language processing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Reference is a central issue for language comprehension and generation. After reviewing existing process models for comprehending and generating referring expressions, we present a general framework for context and reference processing. The context for reference processing is represented as a "Public Workspace." Reference processes access Public Workspace and modify its content, which is the set of concepts currently "on the table" as far as the current language interaction is concerned.

(OVER)

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

407 952

20. (continued)

Information from many different sources can be integrated in comprehending or generating referring expressions. Within this general framework, a new system for selectively generating referring phrases is developed. This system decides how much to express about a given concept in a given context.

ARPA ORDER NO. 2223

ISI/RR-78-72
October 1978

James A. Levin
Neil M. Goldman

Process Models of Reference in Context

INFORMATION SCIENCES INSTITUTE

UNIVERSITY OF SOUTHERN CALIFORNIA

4676 Admiralty Way/ Marina del Rey/California 90291
(213) 822-1511

THIS RESEARCH WAS SUPPORTED BY GRANT MCS76-07332 FROM THE NATIONAL SCIENCE FOUNDATION.
THIS DOCUMENT APPROVED FOR PUBLIC RELEASE AND SALE; DISTRIBUTION IS UNLIMITED.

CONTENTS

ABSTRACT

Reference is a central issue for language comprehension and generation. After reviewing existing process models for comprehending and generating referring expressions, we present a general framework for context and reference processing. The context for reference processing is represented as a "Public Workspace." Reference processes access Public Workspace and modify its content, which is the set of concepts currently "on the table" as far as the current language interaction is concerned. Information from many different sources can be integrated in comprehending or generating referring expressions. Within this general framework, a new system for selectively generating referring phrases is developed. This system decides how much to express about a given concept in a given context.

INTRODUCTION

Wilton Times, W.Va.: "The bad news didn't surprise Miss Ankrom, who is expecting a baby. She said she had been half expecting it."

It is clear that context plays a critical role in the comprehension and generation of language. By itself, the word "it" has little meaning--only in a particular context does it have a referent. Although the importance of context is obvious, models of language processing have not dealt with context in a satisfactory way.

We will discuss some recent process models of the comprehension and the generation of referring phrases. First, we will look at the various representations of context that have been proposed, then we will cover the various ways in which context has been used in the processing of referring phrases, and finally, we will present a model that decides how much to include in the referring phrase generated for a given concept.

THE REPRESENTATION OF CONTEXT FOR REFERENCE PROCESSING

Philosophers of language, linguists, psychologists studying language, and computer scientists building language processing systems have until recently avoided context issues. One reason has been the lack of good representations of context. We will look at some of the process models of language that have used context to deal with referring expressions and then examine some of the psychological evidence of the structure of context in human language processing. Using some of the notions contained in recent representational formats developed in artificial intelligence, we will describe a new proposal for representing context for reference processing.

We are presuming a view of communication in which sequences of words, *referring phrases*, are associated in the minds of the speaker and listener with *referent concepts*. These referent concepts are often also associated in their minds with particular objects and events in the world of discourse, called the *referent objects* of the phrases. Phrases having a common referent concept are called *co-referential*.

Language comprehension requires the discovery of referent concepts for referring phrases, whereas language generation requires the production of referring phrases for referent concepts. The notion of reference is best exemplified by referring phrases which are syntactically noun phrases or pronouns, and by referent concepts which represent concrete individual objects or sets of such objects. However, referring phrases may come in other syntactic forms, and referent concepts may represent events and abstract concepts as well as concrete objects.

Reference, as we are viewing it, is a procedural notion specific to individual process models; one cannot, in general, speak meaningfully of the *correct* referent concept of a given phrase. To illustrate this consider: "I took the paint off the shelf and opened it." One model may decide, say by syntactic means, that the referent of "it" is the paint and then, considering the semantics of "open," decide that what was opened was not the paint itself, but its container. A second model may operate by looking through its context for a "genderless individual" object which can be opened, and find the can of paint as the referent concept of "it." The two models have come to the same interpretation of the sentence, but by different paths. We are distinguishing the question of "what was opened," which relates only to the information content of the sentence, from the question of "what is the referent concept of 'it'," which may have different answers for different processes independent of how they answer the first question.

SYNTAX-DOMINATED CONTEXT REPRESENTATIONS

Though it remains one of the landmark systems for processing natural language, Blocksworld (Winograd, 1972) had a relatively simple notion of context embedded within it. For comprehending and generating referring phrases, particularly pronouns, it relied heavily on searches through the parse trees of the previous utterances. Information about the syntactic position of previously occurring referring phrases stored in these parse trees was used by Blocksworld to find the referent of a referring phrase. This context was dynamic, in the sense that only the few most recent parse trees were searched.

SEMANTIC-DOMINATED CONTEXT REPRESENTATIONS

Another approach to reference processing was taken by the MARGIE system (Rieger, 1975). In this system, the comprehension of a referring phrase was deferred by the parsing system, and was attempted instead by a later inferencing system. As a context within which to process the reference, MARGIE used its memory of "conceptual dependencies." None of the syntactic information about the place of the current expression in the input utterance nor syntactic information about previous expressions was used by the inferencing part of MARGIE. Also, its context was monotonically growing, modified only by addition of conceptual dependency structure derived from the comprehension of successive input utterances.

Several language comprehension models have introduced information schemas which amalgamate knowledge about a central concept (Schank & Abelson, 1975; Bobrow & Winograd, 1977; Hayes, 1977). Once these models have determined that the central concept has been referenced, the entire schema forms a context which makes related concepts available for instantiation. One purpose of this is to focus the search for referent concepts of expressions seen later.

PSYCHOLOGICAL EVIDENCE ABOUT HUMAN CONTEXT

The notion of context that we have been exploring corresponds roughly to the psychological notion of "short-term memory." As the name implies, humans have a memory for events, which goes away after a while. Numerous experiments have investigated the nature of this memory using nonsense syllables, but few have used naturally occurring language, largely because of a lack of an adequate representation for language as a stimulus material. However, in experiments with nonsense syllables, short-term memory has been found to be limited (Miller, 1956) but hierarchically structured (Mandler, 1968).

MULTILEVEL REPRESENTATIONS

Consistent with these findings of hierarchical structure in short-term memory, several recent representational formats have incorporated the notion of multilevel descriptions of concepts. The MERLIN system (Moore & Newell, 1973) and later the KRL system (Bobrow & Winograd, 1977) are representational formats that allow concepts to be represented at many different levels of specificity.

> *El Paso Times*, Texas: "The manager of a nudist park complains that a hole was cut in the wall surrounding the camp. Police are looking into it."

The need for multiple levels of description in reference processing is illustrated by this quotation. Let us consider the issue of determining the referent concept for "it." The humor comes from the incongruity of the interpretations that: 1) the police are looking into the nudist park, or 2) the police are looking into the hole, or 3) the police are looking into the wall. However, note that even when these incongruous interpretations are ruled out, there remain two distinct interpretations: 1) the police are looking into the complaint, or 2) the police are looking into the cutting of a hole. We are not immediately aware of this "ambiguity" because it doesn't really make much difference for the overall interpretation of the sentence. At some level, these two low-level differences in interpretation are covered by a common higher level description.

PROPERTIES OF CONTEXT FOR REFERENCE PROCESSING

To summarize the discussion so far, context for reference processing requires the following properties:

1) it is derived from the language being processed,
2) it draws upon permanent knowledge,
3) it is transient,
4) it is structured,
5) it permits multiple levels of description.

PUBLIC WORKSPACE

Given these desiderata, we have developed a representation for context that is derived from a previous notion of "Workspace" (Levin, 1976) as a kind of structured dynamic short-term memory containing "activations" of permanent "concepts" and relationships between them in long-term memory. Each activation in the Workspace has a numerical salience value; when the salience of a given activation falls below a global threshold level, the activation disappears from the Workspace.

"Public Workspace" is a subset of the overall Workspace, containing only those activations derived from the processing of utterances in the current interaction. In a dialogue, for example, the Public Workspace of a participant contains activations of those concepts and relationships "on the table" as far as the current state of the dialogue is concerned. Since each participant knows that the other participant is currently aware of these concepts, each participant can rely upon this shared awareness in processing referring phrases. This Public Workspace representation is similar functionally to Chafe's "foreground" notion (Chafe, 1974).

THE USE OF CONTEXT IN REFERENCE PROCESSING

So far, we have restricted our discussion of context solely to the representation issues. We are also interested in the kinds of processes that use context to deal with referring phrases. In this section, we will re-examine some natural language systems, looking at the kinds of processes they use, and we will classify these processes into two major categories, which we will relate to our Public Workspace representation of context.

The algorithms used for reference determination are examples of the generate-and-test paradigm. Diverse sorts of information are generally available for finding a referent--syntactic clues, discourse clues, world knowledge--and the algorithms typically use certain sorts of information to *generate* possible referent concepts and other sorts of information as filters to *test* the plausibility of the hypothesized referent concepts.

REFERENCE PROCESSES IN EXISTING SYSTEMS

Reference in Blocksworld

Winograd (1972) implemented heuristics for reference resolution which were able to deal with two modes of reference to the objects in his Blocksworld: reference by partial description and reference by pronoun. For partial descriptions, a list of all blocks that had the properties specified in the description was collected. If the size of this set was appropriate for the quantifiers, qualifiers, and determiners in the description, the set of objects found was used as the referent. If too many were found, an attempt was made to restrict the set to those most recently referenced. If too few were found, the description was reparsed if possible. As a last resort, the system asked for more information.

The process generated possible referents for the pronoun "it" not by using its semantic model, but by applying rule based on syntactic criteria for finding plausible co-referential expressions in the preceding dialogue. The semantic model of the Blocksworld was used to filter out possible referents, and the syntactically most plausible referent concept which passed the semantic filter was accepted.

Reference in MARGIE

Some systems find the referent concepts for expressions solely on the basis of semantic features, ignoring syntactic features. Quillian (1969) described such a system, and Rieger (1975) used a similar approach to find the referent concepts for definite noun phrases within the MARGIE System (Schank, Goldman, Rieger, & Riesbeck, 1973).

In Rieger's approach, a referring expression contributes specifications which the referent concept must satisfy. A search is conducted to find all concepts in memory which are compatible with the specifications. Because the comprehension system need not have the same knowledge about the discourse objects as the speaker has, this compatibility test involves a partial match process, which contributes a score indicating how well an object matches specifications. In addition, Rieger maintained for each concept two attributes which aided in reference resolution. RECENCY, which indicated when the concept last served as the referent concept of some linguistic structure, and TOUCHED, which indicated when the concept was last accessed in reasoning processes. The reference finder used these attributes to define an implicit notion of salience, which was used to help resolve references which the partial match process failed to resolve.

In addition, Rieger's model was able to use its inference capabilities to find specifications not derivable from the referring expression itself. There were two sources of such specifications:

- Any argument of any predicate could have a "specifier program" associated with it. This was a program which could add new specifications for the argument associated with it, contingent on context and on existing specifications for the various arguments of the predicate.

- The general inference mechanism was designed so that it could work with predications mixing particular concepts and described concepts (sets of specifications). When the specifications from the referring expression, together with those added by any applicable specifier program, were insufficient to uniquely identify a referent, those specifications, plus any other features common to *all* remaining candidate referent concepts, formed a temporary concept. The inference rules were then permitted to operate as though this concept were a valid referent. The inference rules might then add further specifications to this concept and eventually narrow the set of candidate referent concepts. If inference processes halted without identifying a definite referent, the temporary concept could be carried along until additional text was processed. In theory, only the need to produce some form of external behavior could force the model to decide on a referent.

CLASSIFICATION OF THE REFERENCE HEURISTICS

Let us look at some reference heuristics, including those we have discussed already, plus additional ones described in papers by Warnock (1972) and Baranofsky (1970).

We can classify them into two categories: those dealing with the possible referent concepts and those dealing with the referring expression.

The heuristics that deal with aspects of the possible referent concepts are a diverse lot, covering both syntax and semantics. One thing seems to be a common feature--each seems to reflect contributions to the salience of the various concepts in awareness. Some of the heuristics capture what is called "focus"; others reflect the fact that concepts in awareness are temporary, disappearing if they aren't refreshed.

1. Proximity: The closer an expression is to the referring expression, the more likely it is to be co-referential with the referring expression. That is, the more recently a concept has been referenced, the more likely it is that the concept is again being referenced.

"I can't find any documentation on the program. I have a tape here at Purdue and I can't figure out what format *it's* in."

The "it" refers to the tape, not to the documentation or to the program.

2. Syntactic role in the sentence:

2a. Subject/Object/Preposition phrases: The referent concept of the syntactic subject of an utterance is more likely to be re-referenced than is the referent concept of syntactic object, which is itself more likely than the referent concept of a prepositional phrase.

"O: The output comes out on the line printer.
L: Throw *it* away . . . "

The "it" refers to the output (the syntactic subject of the first utterance) rather than to the line printer (the syntactic object).

2b. Superordinate/subordinate: Concepts referenced in a superordinate clause are more likely to be re-referenced than those in any subordinate clauses.

" . . . the tape that file is archived on seems to be a bad tape. We can't seem to get *it* to read . . . "

The "it" refers to the tape rather than to the file (in the subordinate relative clause).

2c. Topicalization: Some special syntactic constructions (such as cleft sentences) can be used to emphasize one element of an utterance. The referent concepts of emphasized expressions are more likely to be re-referenced than those of unemphasized ones.

" . . . there is a background job running here that checks to see if there is any unsent mail. Once *it* finds some, *it* tries to resend it."

The two "it"s are co-referential with the topicalized "background job" rather than with the "unsent mail."

3. Centrality: The more frequently a concept has been referenced the more likely it is to be referenced again.

"L: . . . Any chance I can recover [file name] from the most recent system dump?

O: Probably, let me look for *it* and get back to you, ok?

L: Could you SNDMSG to me, one way or the other? I won't be doing anything about *it* tonight. If *it* is there, I will be forever grateful to recover *it*."

The "it"s by L all refer back to L's file, rather than to the system dump or the one way or the other.

4. Current topic: An expression which refers to a concept in the current topic is more likely to be co-referential. Deutsch (1974) observed that definite references can normally be made only to concepts that are part of a currently "open topic." Once a topic is "closed," it must be reopened before concepts within it can be referenced again.

"L: . . . Can you recover those files for me..as far as I know they were in the directory on the 16th. . . the names are . . .

[intervening dialogue]

O: OK I have found *the files you want*. I will retrieve *those* for you . . . "

In the second utterance, O initially specified the files in some detail, but once the topic was re-established, just "those" sufficed.

There are also heuristics that supply information about what the referent of the referring expression can be. These heuristics focus on the information content of the expression and its immediate context.

1. Specification within the referring expression:

1a. The pronoun in an expression often specifies that the referent concept be a particular number and/or gender.

"How do *I* get Runoff to work?"

The pronoun "I" completely specifies the referent concept to be the speaker.

1b. The head noun of the referring expression frequently names a class of objects or events; the referent concept must be an individual within that class.

"I have found *the files* . . . "

1c. The other modifiers in the referring expression, such as adjectives, relative clauses, prepositional phrases, and possessives, often further specify the referent.

"I have found *the files you are concerned about* . . . "

2. Specification by the verb in the same utterance: The verb having the referring expression as an argument often specifies that the referent concept have certain properties.

"Throw *it* away."

The verb (and verb particle) "throw away" specifies the referent to be something of little value to the speaker.

3. Specification by the whole clause containing the referring expression: Sometimes the particular combination of a verb and its noun phrase arguments puts constraints on the referent.

"Did *it* produce any output file?"

The combination of the verb and the object limit the referent to being a computer program of some kind.

A GENERAL FRAMEWORK FOR REFERENCE COMPREHENSION

The classification of the reference heuristics into these two categories suggests a general approach toward modeling reference abilities in process models. There should be two parts of such a model: the processes that contribute specifications to the referent concept of the current expression, and the processes that affect the salience of all the other currently active concepts. Given a spreading activation processing framework (Collins & Loftus, 1975; Levin, 1976; Levin, 1978), we can integrate all of the information supplied by the heuristics concerning aspects of possible referent concepts by having each such process modify the salience levels of the activations in the Public Workspace. Since these changes are global, the resultant salience values are thus the integrated result of all the currently active processes. These salience values can then be used to resolve ambiguous cases, in a way analogous to but more general than the plausibility ratings used by previous systems. With this mechanism, any additional information sources can be added easily by having them modify the salience of activations in the Public Workspace, concurrently with the processes already implemented.

Initially, the referent concept of an expression is completely unspecified. Each of the heuristics concerning the referring expression can be seen as contributing specifications to this unspecified concept. Each heuristic may operate independently in adding its constraints to the referent of the current expression. The end result is the partially specified referent.

ELLIPSIS

An issue closely related to reference determination is that of ellipsis, a surface form which omits the specification of certain syntactic or semantic roles. Most models of language comprehension attempt to "recover" omitted material in at least some situations, that is, to find the referent concept of an "empty" expression.

A linguistic approach to this problem is to view ellipsis as the result of deletion rules in a grammar, and to recover the omitted concept by reconstructing the parse tree from which the deletion was made. The result is to determine that the empty expression is co-referential with some expression appearing in the surface structure, to which the normal rules for reference determination can apply.

The grammar-based approach offers no solution to many forms of ellipsis, however. For instance, it cannot determine the missing object of "fired" in

The soldier picked up his rifle. He fired <empty>.

A straightforward approach to finding the referent in such a case is to treat <empty> as a numberless, genderless pronoun and apply available rules for finding the referent concepts of pronouns. It is not clear, though, when grammatical rules for pronouns can be extended to omitted arguments, or to what extent the observed probabilities for locations of antecedents extend to cases of ellipsis.

One reason that process models attempt to find or create referent concepts for omitted material is that these referent concepts may themselves be the referent concepts of expressions appearing later in a text:

The soldier took careful aim and fired.
The rifle's recoil knocked him down.

Both the proximity and syntactic role rules are enhanced by the recovery of omitted arguments.

It has also been noted that once a concept has been referenced, certain concepts predictably associated with that concept can serve as referent concepts of later expressions without any explicitly mentioned link to the first concept. This form of reference has been handled, at least in some cases, by the introduction of knowledge schemas, as previously described. When a text presents sufficient evidence to "instantiate" one of these units, various concepts within the instantiation become available as potential referent concepts, even though they have never been explicitly mentioned in the text. Using a "script" about restaurants, SAM (Schank, 1975) can process:

John went to a restaurant. The hostess seated John.

and interpret it in a way that connects the referent concepts of "restaurant" and "hostess" in a rational manner.

REFERENCE AND GENERATION

In comparison with the large investment in process models of language comprehension, little study has been devoted to the process of language generation. This is particularly true with respect to reference. The reference problem for a generator can be stated as

> Given the goal of expressing something about a concept C in a particular context, how should C be referred to? That is, what subset of the information known about C should be expressed and *how* should it be expressed?

It is possible to generate phrases to refer to concepts of all sorts: specific physical objects, events, abstract qualities, generic objects, sets, predicates, etc. The natural language generator has two orthogonal concerns. It must determine what information to convey about the concept, and it must select the words and syntactic forms appropriate to carry out the reference. The choice of information does not necessarily dictate the words and syntax:

> I saw *the Ford and Chevy collide.*
> I saw *the collision between the Ford and the Chevy.*

Nor does a choice of syntax necessarily dictate words or information content.

> I saw the *collision between the Ford and the Chevy.*
> I saw the *automobile crash.*

The generator not only must produce grammatical referring phrases for individual concepts, but must maintain grammaticality when those phrases are combined. Certain choices may constrain others. For example, "I assume the Ford and Chevy collided" is grammatical, while "I assume the collision between the Ford and Chevy" is not.

In generating a referring expression, a generator may make use of the previously described aspects of such expressions: explicit specifications within the phrase, specifications implicit in the syntactic role played by the phrase with respect to a verb, and specifications implicit in a whole clause. This means that the generator may be able to make tradeoffs within a sentence; the selection of a particular verb or the inclusion of certain specifications in one referring expression may affect the information content needs of another reference.

It is possible for a verb or phrase to contribute implicit specifications to a preceding phrase, so a strictly left-to-right generation strategy cannot account for some reasonable references. Consider, for example, the common form of solitaire in which a legal move consists of placing a card onto another card of opposite color. In a situation with one king and one queen of each color available, the statement "Place the queen on the king" is ambiguous, whereas both "Place the red queen on the king" and "Place the queen on the black king" unambiguously specify the same move.

EXISTING REFERENCE GENERATION MODELS

Two generation models (Simmons & Slocum, 1972; Goldman, 1975) addressed only the issue of how to express as an English string the information that they were given. Both presumed that no information selection was required. Blocksworld (Winograd, 1972) was designed to carry on a dialogue and thus had to deal with the selection aspect of reference generation. The task domain allowed three simplifying assumptions:

- All information known to the generator is also known to the listener. Thus, any object which needs to be referenced can be referenced by selecting sufficient information *distinguishing* it from other objects. Furthermore, any piece of information which can discriminate between objects in the robot's model will enable the listener to perform the same discrimination.

- The number of objects in the world model at any time is sufficiently small that it is computationally feasible to determine whether a description uniquely describes an object. Thus, it is not necessary to maintain a context, or submodel, within which to make the discrimination.

- The information about each object is sufficiently small, and sufficiently regular, that the selection process can avoid search. In Blocksworld, this amounted to selecting properties in a fixed order until the accumulated properties uniquely described the object.

Winograd's program produces both definite and indefinite noun phrases to refer to toy blocks. The choice between definite and indefinite is made on the bases of syntactic features of the question being answered. The noun phrase is constructed by concatenating adjectives naming the block's size and color with a noun naming its shape. In cases where a definite reference is required and size, shape, and color are insufficient to uniquely identify the block being referenced, relative clauses can be added which give the block's proper name, if it has one, and its spatial relationship to other blocks.

A few discourse heuristics in Blocksworld affected the generation of referring expressions. Sets of individual blocks were normally referred to by conjoining the descriptions of the individuals in the set. However, if the descriptions were identical, which could happen when indefinite descriptions were used, the entire set was described by prefixing its cardinality to a description of one of the elements and making the noun plural--"three small cubes." Pronouns would be introduced if the concept being referenced had served as the referent of a "nearby" expression.

A SYSTEM FOR THE SELECTIVE GENERATION OF REFERRING PHRASES

In this section, we will describe a system we have developed that uses the notions described in the previous sections to tackle some of the problems of deciding how much to say about a given concept. In a typical generation task, we want to express a given concept in service of some particular generation goal. But there is then an issue of how much needs to be said about the concept. It is almost never appropriate to express everything known about the concept. Obviously, the choice of what to say will depend upon the context within which the referring phrase will exist.

Grice (1975) has specified at a very general level some of the rules for deciding what to say, in a form called "conversational postulates." Of particular concern for us is his "Postulate of Quantity":

1. Make your contribution as informative as is required (for the current purposes of the exchange).

2. Do not make your contribution more informative than is required. (Grice, 1975, p. 45)

We have implemented a system that, given a particular concept to express and a particular context, creates recommendations of what aspects of the concept to express and what aspects not to express. This system, called CES, represents context as a Public Workspace, as already described, and contains processes for using the salience of the activations in Public Workspace and the specifications of the current concept.

WHAT TO SAY

Consider a particular concept to express. In the movie *The Tall Blond Man with One Black Shoe* there is a character who is a tall blond man with one black shoe (and one brown shoe). Each time that we talk about this man, we don't want to have to mention all of these specifications; yet we usually have to mention at least some of them. This is about as far as the Postulate of Quantity take us, which isn't far enough to implement a process model of deciding what to say. We have to know in a bit more detail how to determine what is just enough to say, not too little or too much.

Suppose the context is set by the comprehension of the sentences "The tall blond man with one black shoe was unarmed. He was chased by the handsome dark haired spy with a golden gun." Within the framework for reference presented here, this means that the Public Workspace of the Generator (and, presumably, also the Hearer) contains an activation of the blond, with specifications of tall, blond, man, having one black shoe, and being unarmed, and also an activation of the spy, with specifications of handsome, dark haired, spy, and having a golden gun. Suppose that we now want to generate a sentence saying that the dark haired spy slipped. We can't just say "He slipped" because "he" doesn't distinguish between the tall blond and the spy. It would be overkill to say "The handsome dark haired spy with the golden gun slipped." A marginal expression in this context is "The handsome man slipped," since we don't know whether the tall blond is handsome or not. A better expression is "The dark haired man slipped." Here we start to see some of the factors determining the selection process--we want to say at least enough to distinguish the concept we want to convey from the others in Public Workspace. This is natural if we consider the purpose of generating the expression: to "stimulate" an activation of this concept in the workspace of the Hearer. This goal can be achieved only if we supply enough information to rule out the other concepts currently "on the table" in the conversation.

Consider another sentence we might want to generate within this same context. We want to say that the spy threatened to shoot the blond. In this case, the comprehender of the verb "shoot" in this sentence can generate an expectation that the person shooting has a gun. So it is not necessary to express the specifications that the spy had a gun. Moreover, we can use this expectation (and any others we have for the position in the surface sentence where the phrase will appear) to rule out possible ambiguities, even *before* anything is expressed. Thus, in this case, we only need to rule out the inanimate objects to convey the spy--saying "he" is sufficient.

In addition, the first part of the sentence "He threatened to shoot . . ." generates expectations for the other parameter of "shoot" that uniquely specify that the tall blond is intended, even before *anything* is said. So we need not say anything: "He threatened to shoot." In this way, we can generate ellipsis in cases allowed by syntactic considerations.

The CES System recommends expressing the specifications of the current expression that rule out any "confusion candidates" in the Public Workspace. A confusion candidate is an activation in the Public Workspace that is consistent with everything known so far about the current expression. This knowledge includes the expectations about the current position that the hearer could be reasonably expected to share, and the specifications so far recommended for expression.

Expectations

The expectations about the current position may be generated by the other language generation processes running concurrently with the CES component. For example, expectations can be generated about the current position from multisentential structure processes, like Scripts (Schank & Abelson, 1975) or the Dialogue-Game Processor (Levin & Moore, 1977). Currently, however, CES collects expectations from only two sources. First, it looks at the predicate of which the current concept is a parameter. In Long-Term Memory, there may be general information known about this parameter position which can then be expected to hold about the current concept. These expectations, plus those directly supplied by the calling process, form the expectations about the current position.

Order of Selection

In the process of considering which specifications to express, we clearly want to select those that rule out at least one of the confusion candidates. However, there are many different possible sets of specifications, each of which would rule out all the confusion candidates. Finding the "optimal" (smallest) set would be difficult, introducing a combinatoric explosion problem. For the CES system, we have selected an approximation to an optimal choice. At each point in the process, the CES system selects the specification to express that rules out the largest number of confusion candidates. This process continues until all confusion candidates are eliminated.

What NOT to say

The CES system decides that a specification need not be expressed for one of two different reasons. First of all, we have the expectations about the current position, described earlier. Since these are generated from common knowledge, the hearer probably shares these expectations. So, any specification of the current concept that is in this expectation set is marked by CES as not to be expressed, since it would be redundant to express it.

The other way that CES decides what not to say occurs later in the processing of a given concept. CES recommends expressing specifications until all of the confusion candidates are ruled out. Those remaining need not be expressed.

Example

Let us run through a quick example of how the CES system operates. Given the context described previously for the tall blond man and the spy, how does CES operate to decide how to express the handsome spy slipping? First, expectations are collected from the general concept for "slip." If the agent were constrained to be animate, CES will use this constraint as an expectation on this agent. CES now will rule out all inanimate concepts, in this case, the golden gun and the black shoe. To handle the case of expectation violation, as in metaphorical uses of language, if an expectation rules out the concept to be conveyed, CES will ignore the expectation. But CES still has to eliminate the remaining confusion candidate, the tall blond man. The specification of dark-haired is selected, because it explicitly rules out the blond man.

In a more complex case, CES will try each specification to see how many of the confusion set are ruled out, and recommend expressing the one that rules out the most. This cycle continues until no more confusion candidates exist, or no more specifications exist.

In some cases, CES will be unable to rule out one or more confusion candidates, even after considering all known specification of the current concept. Currently, CES gives up and admits its inability to uniquely specify the concept. Future directions for research may deal with these problem cases, perhaps by use of text reference ("the second spy I mentioned"), or the introduction of nonrestrictive specifications, i.e., those in the generator's private Workspace but not in the Public Workspace.

SUMMARY

We have presented a framework for reference processing, derived from a taxonomy of the various heuristics currently used to comprehend references. Within this general framework, we described a model for selecting what to say (and what not to say) when generating a referring phrase for a concept. The framework proposes that the various sources of information bearing on reference processing are integrated by having the separate processes all producing global effects on one common Public Workspace. In this way, many different processes can be defined that operate autonomously to contribute information of differing degrees of certainty, which are integrated by the general framework presented here.

Our model of referring phrase generation has been partially implemented and tested on several examples for a simple data base. We believe that the model cannot be assessed adequately until an implementation has been integrated into a text generation process capable of maintaining the Public Workspace data base across multiple utterances.

REFERENCES

1. Archbold, A. A. 1975. Text reference and repeated propositional reference: Concepts and detection procedures. Unpublished paper, Information Sciences Institute, August 1975.

2. Baranofsky, S. 1970. Some heuristics for automatic detection and resolution of anaphora in discourse. Unpublished master's thesis, University of Texas, 1970.

3. Bobrow, D. G., and T. Winograd 1977. "An overview of KRL, a knowledge representation language," *Cognitive Science*, 1, 3-46.

4. Chafe, W. L. 1972. "Discourse structure and human knowledge," in R. O. Freedle and J. B. Carroll (eds.), *Language Comprehension and the Acquisition of Knowledge*. Washington, D.C.: Winston.

5. Collins, A. M., and M. R. Quillian 1972. "How to make a language user," in E. Tulving and W. Donaldson (eds.), *Organization of Memory*. New York: Academic Press.

6. Deutsch, B. G. 1974. "The structure of task oriented dialogs," *Proceedings of the IEEE Symposium on Speech Recognition*, Pittsburgh, Pa.

7. Goldman, N. 1975. "Conceptual generation," in Schank, R.C. (ed.), *Conceptual Information Processing*. New York: American Elsevier.

8. Grice, H. P. 1975. "Logic and conversation," in P. Cole and J. L. Morgan (eds.), *Syntax and Semantics: Speech Acts*, Vol. 3. New York: Academic Press.

9. Hayes, P. 1977. "On semantic nets, frames, and associations," *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, Cambridge, Mass.

10. Levin, J. A. 1976. Proteus: An activation framework for cognitive process models. Doctoral dissertation, University of California, San Diego.

11. Levin, J. A. 1978. Continuous processing with a discrete memory representation. Paper in preparation, Department of Psychology, University of California, San Diego.

12. Levin, J. A., and J. A. Moore 1977. "Dialogue games: Meta-communication structures for natural language interaction," *Cognitive Science*, 1,4.

13. Mandler, G. 1968. "Association and organization: Facts, fancies, and theories," In R. R. Dixon and D. L. Horton (eds.), *Verbal Behavior and General Behavior Theory*. Englewood Cliffs, N.J.: Prentice-Hall.

14. Miller, G. A. 1956. "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *Psychological Review*, 63, 81-97.

15. Moore, J., and A. Newell 1973. "How can MERLIN understand?" in L. Gregg (ed.), *Knowledge and Cognition*. Hillsdale, N.J.: Lawrence Erlbaum Assoc.

16. Nash-Webber, B. 1978. "Syntax beyond the sentence: Anaphora," in Spiro, et al. (eds.), *Theoretical Issues in Reading Comprehension*. Hillsdale, N. J.: Lawrence Erlbaum Assoc. (forthcoming).

17. Quillian, M. R. 1969. "The teachable language comprehender: A simulation program and theory of language," *Communications of the ACM*, 12, 459-476.

18. Rieger, C. 1975. "Conceptual memory and inference," in Schank, R.C. (ed.), *Conceptual Information Processing*. New York: American Elsevier.

19. Rumelhart, D. E., and J. A. Levin 1975. "A language comprehension system," in D. A. Norman, D. E. Rumelhart, and the LNR Research Group, *Explorations in Cognition*. San Francisco: W. H. Freeman.

20. Schank, R. C. 1975. "Using knowledge to understand," in R. Schank and B. L. Nash-Webber (eds.), *Theoretical Issues in Natural Language Processing*. Cambridge, Mass.: Bolt Beranek and Newman.

21. Schank, R. C., and R. P. Abelson 1975. "Scripts, plans, and knowledge," *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*, Tbilisi, USSR.

22. Simmons, R., and J. Slocum 1972. "Generating English discourse from semantic networks," *Communications of the ACM*, 15,10.

23. Warnock, E. H. 1972. Anaphoric reference: Finding the antecedent. Unpublished paper, Harvard University, 1972.

24. Winograd, T. 1972. "Understanding natural language," *Cognitive Psychology*, 3, 1-191.