PROCEDURES FOR INTEGRATING KNOWLEDGE IN A SPEECH UNDERSTANDING SYSTEM

Donald E Walker and William H Paxton, with
Barbara J. Grosz Gary G. Hendrlx, Ann E Robinson, Jane J. Robinson and Jonathan Slocum
SRI International Menlo Park, California 94025

This paper describes the procedures for integrating knowledge from different sources in the SRI speech understanding system. A language definition system coordinates—at the phrase level —information from syntax, semantics and discourse in the course of the interpretation of an utterance. The system executive uses these contextual constraints in assigning priorities to alternative interpretations, combining top-down, bottom-up, and bidirectional strategies as required Experimental results that demonstrate the effectiveness of context checking are discussed.

Descriptive Terms: Language Understanding> Speech Understanding, System Control, Language Definition, Syntax, Semantics, Discourse

1 INTRODUCTION

For the past five years, SRI International has participated in a major program of research on the analysis of continuous speech by computer, sponsored by the Advanced Research Projects Agency of the Department of Defense. The goal of the program was the development of a speech understanding system capable of engaging a human operator in a natural conversation about the performance of a particular task (see Newell et al ., 1973). A rather complex set of specifications defined the parameters more precisely. The program culminated in the demonstration of a system that did meet the target specifications (see Reddy et al., 1976; Medress et al., 1977), but more important for artificial intelligence were developments in the various knowledge sources—particularly syntax, semantics, and discourse—and in the system architecture necessary for coordinating them efficiently and effectively.

At SRI, we have made significant advances both in building components for knowledge sources and in developing a' framework for coordinating and controlling them. The syntactic component is a performance grammar; it describes the syntax of the English used in spontaneous dialog rather than the English of edited text. The semantics component uses a partitioned semantic network to enable the representation of multiple alternative parses without duplication, the association of syntactic units with their network images and the establishment of scopes for higher order predicates (especially quantifiers). A discourse component has been developed that uses the context of the preceding dialog to resolve definite noun phrase references and expand elliptical utterances; it works on the partial knowledge available when parsing an utterance.

This research has been funded under the following ARPA contracts, all administered through the Army Research Office: DAHC04-72-C-0009, DAHC04-75-C-0006, and DAAG29-76-C-0011.

Our approach to coordination and control stresses integration—the process of forming a unified system out of the collection of components --and control—the dynamic direction of the overall activity *of* the system during the processing of an input utterance. Our approach to integration provides for interactions of information from various sources of knowledge in a procedural representation; a means for adjusting the language definition to particular domains without loss of generality; and avoids commitment to a particular system control strategy, thus allowing flexibility in combining words and phrases. Our approach to control provides special techniques to assign priorities by using contextual constraints; allows combinations of top-down, bottom up, and bidirectional strategies; organizes and constructs data structures *for* hypotheses in a manner that greatly reduces duplication of effort; and is based on extensive experimental studies to evaluate design alternatives.

A review of the total project is beyond the scope of this paper Instead we will concentrate on integration. After presenting an overview of the operation of the system to provide context, we will present brief descriptions of the knowledge source components mentioned above together with more extensive discussions of a facility for language definition that provides the basis for coordinating them and of the executive routines that control them. We conclude with a consideration of relevant experimental results.

A more complete statement of this work is contained in our final project report (Walker, 1976). A somewhat expanded description of the language definition system and executive and the experiments conducted to test them is presented in Paxton (1977; sec also 1976a,b). The discourse component is treated more fully in Grosz (1977a; see also 1977b for a discussion of the. concept of focus). Fikes and Hendrlx (1977) summarize the scheme for semantic representation and the procedures for deductive retrieval used in the system References to other papers are included in the final project report.

The system referenced in this paper was developed jointly by SRI and the System Development Corporation (SDC). SRI provided capabilities for system control, language definition, syntax, semantics, and discourse analysis. SDC provided capabilities for signal processing, acoustics, phonetics, and phonology, as well as special system software and hardware support (see Bernstein, 1975). The task domain supplied data management capabilities for querying a file of information on attributes of ships from the U.S., Soviet, and British fleets.

Early in 1976, just after the system became operational, the computer facility at SDC was removed and our collaborative efforts stopped. Although we were not able to exercise the system extensively or refine the interface between the

acoustic components and those providing higher level language processing, we were able to collect data on the performance of the SDC components and have used this information in extensive tests of our components and of the system framework. It is on the basis of these experiments that we report on the SRI developments in speech understanding in this paper

II THE *OPERATION*

When a speaker records an utterance, it is analyzed acoustically and phonetically, and the results are stored in a file. When these data are available, the executive begins to predict words and phrases on the basis of context—guided by the rules for phrase formation in the language definition—or to build up phrases from words that have been identified acoustically in the utterance. As each phrase is constructed, relevant semantic and discourse information is checked, and if appropriate, a semantic network representation of the phrase is developed. When the performance of a task results in the prediction of a word at a specified place in the utterance, alternative phonological forms of that word are mapped onto the acoustic data for that place, and a score indicating the degree of correspondence is returned. When an interpretation for the entire utterance is complete, relevant structures from the semantic model of the domain and from an associated relational data base are processed to identify in semantic network form the content of an appropriate response. This response is then generated either in text form or through the use of a speech synthesizer.

III THE LANGUAGE DEFINITION

The input language is a subset of natural, colloquial English that is suitable for carrying on a dialog between a user and the system regarding information in the data base. The definition of this language consists of a lexicon containing the vocabulary, a set of composition rules for combining words into phrases and smaller phrases into larger ones, and some global declarations giving information needed by the definition compiler, which translates the language definition into an efficient internal representation, and by the executive, which operates on that representation. The lexicon is separated into categories, such as noun and verb, and the words in each category are assigned values for various attributes, such as particular grammatical features and semantic representations. The composition rules are phrase-structure rules augmented by a procedure that is executed whenever the rule constructs a phrase. Information provided by the procedure includes both attributes of the phrase based on the attributes of its constituents, and factors for use in judging the acceptability of the phrase. The global declarations give such information as lists of attributes for the different categories.

An attribute statement may compute values that specify acoustic properties related to the input signal, syntactic properties such as mood and number (singular or plural), semantic properties such as the semantic network representation of the meaning of the phrase, and discourse properties for anaphora and ellipsis. The values of constituent attributes are used in computing the attributes of larger phrases, and the attributes of complete interpretations are used in generating responses.

The factor statements compute acceptability ratings for an instance of the phrase. The scores for factors are non-Boolean; that is, they may assume a wide range of values. As a result, a proposed instance of a phrase is not necessarily simply accepted or rejected; it may be rated as more or less acceptable depending on a combination of factor values. Like attributes, factors may be acoustic, syntactic, semantic, or discourse related. Acoustic factors reflect how well the words match the actual input; syntactic factors deal with tests like number agreement between various constituents; semantic factors assure that the meaning of the phrase is reasonable; and discourse factors indicate whether an elliptical or anaphoric phrase makes sense in the given dialog context. The values of factors are included in a composite score for the phrase. The scores of constituents are combined with the factor scores to produce the scores of larger phrases, and the scores of complete interpretations are used in setting executive priorities.

The attribute and factor statements in the procedural parts of the rules contain specifications for most of the potential interactions among system components. Attributes and factors either have constant values or have values that depend on attributes of constituents and global information (such as a model of the discourse or the results of preliminary, low-level acoustic processing). By design, the attributes and factors of a phrase are not allowed to depend on the context formed by other phrases that can combine with it to produce larger structures. By giving up explicit context dependency, it becomes easier to share phrases among different contexts, which allows the executive to reduce duplication of effort. However, contextual restrictions provide heuristic information that in practice is too valuable to ignore, so the executive algorithms for priority setting take them into account by special techniques described below. Essentially, we have taken the explicit context dependencies out of the rules so that phrases can be shared by different contexts, and have developed methods in the executive so that contextual restrictions can still be used in controlling the operation of the system.

The form of the rules is designed to avoid commitments to particular system control strategies. For example, the rule procedures can be executed with any subset of constituents, so incomplete phrases can be constructed to provide intermediate results, and it is not necessary to acquire constituents in a strictly left-to-right order This flexibility has made it possible to experiment with alternative system control strategies.

IV SYNTAX

The syntactic knowledge in the system is represented both in the phrase structure part of the language definition rules and in the attribute and factor statements in the procedural part of the

rules. Syntax provides computationally inexpensive information about which words or phrases may combine and how well they go together. In testing word or phrase combinations, syntactic information alone often can reject an incorrect phrase without requiring costly semantic and discourse analysis Factors are used for traditional syntactic tests such as agreement for person or number, but factors also are used to reduce the scores of unexpected phrases. For example, WH questions that are negative (e.g. "What submarine doesn't the U.S own?") are not expected to occur. A factor statement lowers the value for this interpretation but does not eliminate it completely, so that if *no* better hypothesis can be formed to account for the input utterance, this interpretation will be accepted. Since the language definition system provides the capability for evaluating phrases in context by means of non-Boolean factors, the grammar can be tuned to particular discourse situations and language users simply by adjusting factors that enhance or diminish the acceptability of particular interpretations. It is not necessary to rewrite the language definition for each new problem domain

The constituent structures defined by the composition rules in the system allow WH questions, How Many questions. How+Adiective questions, imperatives and statements; BE and DO verb forms are included. Noun phrases can be of a variety of types. We have concentrated on those relevant for the domain having WH determiners (what, which, whose), quantifiers (all. any, both, each either, every, neither no, none, some), partitive expressions (containing "of") expressions with numbers (from one through the millions) and units (tons, feet, knots), and comparisons involving numbers. There are more than 100 different kinds of basic noun phrases, to which may be added recursively "of NP" expressions and some classes of prepositional phrases. The lexicon contains over 600 entries; regular plurals, past and past participle forms and other suffixes are handled by rules.

V SEMANTICS

The system's knowledge about the domain is embodied in-a partitioned semantic network. A semantic network consists of a collection of nodes and arcs where each node represents an object (a physical object, situation, event, set, or the like) and each arc represents a binary relationship. The structure of our network differs from that of conventional networks in that nodes and arcs are partitioned into spaces. These spaces, playing in networks a role roughly analogous to that played by parentheses in logical notation, group information into bundles that help to condense and organize the network's knowledge. Network partitioning serves a variety of purposes in the speech understanding system: encoding logical connectives and higher-order predicates, especially quantifiers; associating syntactic units with their network images; interrelating new inputs with previous network knowledge while maintaining a definite boundary between the new and the old; simultaneously encoding in one network structure multiple hypotheses concerning alternative incorporations of a given constituent into larger phrases; sharing network representations among competing hypotheses; maintaining Intermediate results during the question-answering process; and defining hierarchies of local contexts for discourse analysis.

The network includes an encoding of knowledge about the domain of discourse. This model serves as a foundation on which the structures corresponding to new utterances are built; it is used to assess the feasibility of combining utterance constituents to form larger phrases; and it is a source of information for answering queries, supplemented by a relational data base which can be accessed directly from the network.

Concepts in the domain that can be referenced by individual words are so listed In the lexicon. As lexical items are combined into phrases, these network references are passed to semantic composition routines. These routines construct new network structures that represent the meanings of the composite phrase, encoding new instances *or* new combinations of concepts. As phrases are combined into larger phrase units the composition routines are applied until an interpretation for the entire utterance has been constructed In this process, case relations provide a basis for using syntactic information in constructing semantic interpretations; they also allow rejecting unallowable semantic structures and blocking syntactic predictions for words that cannot fit In the current context. The interpretation for the entire utterance takes the form of a network fragment anchored to concepts in the original domain model, but maintained in a separate partition called a scratch Space

The scoping of quantifiers, contained implicitly or explicitly in the utterance, is performed in a separate step after an interpretation has been assigned to the total input. Scoping is accomplished by adding new partitioning to the network fragment without changing the topology of the existing structure.

Once a network structure encoding a fully quantified interpretation of an Input is formulated, it is passed to a response component. For questions and commands requesting information, calls to a deduction component are generated to retrieve or derive the requested information from the network encoding of the domain model.

VI DISCOURSE

The discourse component of the speech understanding system relates a given utterance (or a portion of it) to the overall dialog context and to entities and structures in the domain. The procedures we have developed are based on systematic studies of dialogs between two people performing some activity together. Contextual influences were found to operate on two different levels in a discourse. The global context—the total discourse and situational setting—provides one set of constraints on the interpretation of an utterance. These constraints are used in identifying the referents of definite noun phrases. The second set of constraints is provided by the

See Walker, 1976, and FIkes and Hendrix, 1977, for a description of the deduction component.

immediate context of closely preceding utterances. These constraints are used in the interpretation of elliptical expressions to expand utterance fragments into complete utterances. For the data base domain of the speech understanding system, the discourse context is limited to a linear history of preceding interactions For complex task-oriented dialogs the linear discourse history can be replaced by a structured history related to the organization of the task being performed (see Grosz 1977a,b).

To determine the referents of definite noun phrases, it is necessary to group those parts of the global knowledge base (that is the knowledge about the domain of discourse) that are in the focus OF attention of the dialog participants. To encode this focus, we have introduced a special partitioning that is independent of the logical partitioning used to represent semantic information. Spaces in the focus partitioning are U3ed to group together and highlight those items in the knowledge base that are relevant at a given point in a dialog.

The central process necessary for resolving definite noun phrases is finding a network structure that matches the semantic structure built for the noun phrase. The matching is performed by the deduction component, which associates nodes and arcs in the network fragment corresponding to the noun phrase with nodes and arcs in the knowledge base. The focus representation is used to constrain the search required by this matching process on the basis of the discourse context.

The constituents missing from an elliptical utterance are found in the interpretation of the immediately preceding utterance. The process of building an interpretation for an elliptical phrase entails two steps. First, the items missing from the current utterance are found in the interpretation of the preceding utterance (or, equlvalently, the slot that the elliptical phrase fills in the preceding utterance is determined) Syntactic information plays a major role in this identification, because the corresponding elements usually are structural units of the same type. Semantic closeness, determined from the taxonomic hierarchy of the network, also is considered. Second, a complete phrase is built using the elliptical phrase and the missing constituents found in the previous utterance. The coordination of syntactic and semantic information, achieved through network partitioning, is used to minimize the- computation required.

VII EXECUTIVE

The executive has three main responsibilities: (1) it coordinates the work of the other components of the system by calling acoustic processes and applying language definition rules; (2) it assigns priorities to the various tasks in the system; and (3) it organizes hypotheses and results so that information is shared and duplication of effort is avoided. When a successful interpretation has been found, the executive invokes the response functions, which produce a reply.

The principal data structure used by the executive is called the parse net. It is a network with two types of nodes phrases and predictions. Phrases correspond to words or composition rules from the language definition; phrases can be complete, containing all their constituents, or incomplete, with some or all of their constituents missing A prediction is for a particular category of phrase associated with a particular location in the utterance. As the interpretation of an utterance progresses, new phrases that have been constructed from existing phrases or from words found in the utterance are added to the parse net. At the same time, new predictions are made as more information is obtained. Thus, as the interpretation process advances, the parse net. which holds intermediate hypotheses and results, grows. A complete root category phrase (typically, a sentence) with its attributes and factors constitutes an interpretation of the utterance

There are two tasks entailed in maintaining and evolving this parse net: the word task and the predict task The role of the word task is to look for a particular word in a particular location in the utterance. If the acoustic mapper has not been called previously to test for that word in that location, the word task calls it. If a word is found successfully in the specified location the word is used to build new phrases.

The role of the predict task is to make a prediction for a word or phrase that can help complete an incomplete phrase. Whenever a new constituent is inserted into an incomplete phrase, any adjacent constituents that had been missing can be predicted. New predictions can include predictions for particular words, leading to new instances of calls on the word task.

A prediction serves as an intermediary between two sets of incomplete phrases* consumer phrases that all are missing a constituent of the predicted category at the predicted location in the input, and producer phrases that all might supply the missing constituents Note that a phrase can be a consumer for one prediction and a producer for another. The full set of producer-consumer connections in the parse net makes explicit the different sentential contexts for each phrase. This contextual information is used by the executive in setting priorities and in lookahead.

Establishing the priority of a task begins with determining the score of the phrase involved. The score is computed from the results of the acoustic mapping of any of the words contained in the phrase, from the factor statements for the phrase, and from the scores of the constituents. The score is thus a local, context-free piece of information about how good the phrase is. After the score Is determined, the phrase is given a rating that is an estimate of the best score for an interpretation that can be constructed for a phrase of the root (sentence) category that uses the given phrase. The rating for a phrase does depend on its consumers, that is, on the other phrases in which it may be embedded to form a sentence. This rating is then modified depending on the control strategy being used, and the result is the priority of the task to be performed for that phrase.

The design of the parse net was inspired by Kaplan's (1973) multi-processing consumer-producer approach.

Both the word and the predict task can work either left-to-right through an input or bidirectionally from words selected at arbitrary positions within an utterance. The system is designed to allow constituents of phrases to be added in any order, so experimentation with a variety of control strategies has been possible. More importantly from the system control standpoint, each task does a limited amount of processing and then stops after scheduling further operations for later. The scheduling does not specify a particular time for a future operation but instead gives the operation a certain priority. The operation is performed when it becomes top priority. This organization allows the executive to control the overall activity of the system by setting task priorities.

VIII ENTAL RESULTS

We have experimented with two techniques for using the consumer context in setting phrase ratings In one. called the merging method, the rating with respect to a particular consumer is formed by adding the phrase score and the consumer rating. (Whenever possible, ratings are assigned top-down in the parse net so that consumer ratings are directly available for use in this process) The phrase rating is then the maximum rating with respect to any of its consumers. This method is fast, but it leaves the rating unaffected by the consumer restrictions that are expressed in rule procedures rather than in structure declarations. A phrase may satisfy the structural requirements of a consumer C but still be incompatible with C because of constraints encoded in C s factor statements. For example, if the only sentential context being considered is "Is it owned by —", the structural requirements will be satisfied by any noun phrase, but semantic factors will restrict the alternatives to possible owners.

The second technique for setting phrase ratings is called the context-checking method. It takes into account the procedural information in the rules by exploring paths in the parse net and executing the corresponding procedures to gather attribute and factor information. Each producer-consumer path from a phrase P to a root category phrase reflects a way of constructing an interpretation using P. Various paths from P are formed, and the rating for P is its best rating with respect to any of the constructed paths.

To reduce the cost of rating alternatives by the context-checking method, a heuristic search is made in the parse net for a near optimal path rather than exhaustively trying all possibilities. The heuristic exploits the fact that, typically, when a phrase is being rated, the higher level phrases that form its context have already been rated. (The parse net is initialized so that a context of previously rated phrases exists even when the system is doing bottom-up processing.) These prior ratings provide important heuristic information. The object is to find the path giving the best score, so the paths with the highest prior rating are explored first. When a complete path is found, one that leads to a root-category phrase, the score for that path is used as a threshold to prune other paths that have lower expected ratings.

The context-checking method takes more computation per rating assignment than the merging method but it produces better phrase ratings since it gathers more information in forming them. The experimental results support the conclusion that the extra effort spent in this method is worthwhile; It leads to better system performance as reflected in both accuracy and runtime As a background for consideration of these results it is necessary to consider briefly the scope of the experimental effort as a whole.

The experiments to determine the effects of variations in control strategy for our speech understanding system were conducted using a simulated version of the acoustic processing components as indicated above Although we lost access to the mapper as a result of the removal of the computer facility on which the total system was being implemented, there are compelling reasons for doing simulation experiments. Extensive testing with the actual mapper would have been impossible both because of the time required and because of increased demands on memory with the large delays for page swapping by the time sharing system. In addition we were able to study control strategies that otherwise would have been too slow for the mapper

The mapper simulation reproduces the observed mapper performance statistics for the false-alarm rate the hit scores and the particular false-alarm words and their scores Because of insufficient data, we were not able to simulate more complex statistics such as the cooccurrence of hits and false alarms or the dependence of scores on position within the utterance. Consequently, the experiments are designed to emphasize comparisons between performance levels for the different design alternatives and should not be taken as estimates of the absolute value of the system's performance with a real mapper For this reason we do not consider here the effectiveness of the system as a whole, although there are observations that are relevant for an evaluation of the overall results of the ARPA Speech Understanding Research Program (see the sections on Experimental Studies in Paxton 1977, or Walker, 1976).

In the main experiment of the series, the performance of the speech understanding system, with a lexicon of 305 words, was measured on a set of 60 test sentences, while varying four major control-strategy design choices. The sentences covered a wide range of vocabulary and included questions, commands, and elliptical sentences. The 60 sentences ranged in length from 0.8 to 2.3 seconds. There were 10 sentences at each 0.3 second interval. The sentences averaged 5.9 words in length, with a maximum of 9 words.

In addition to studying the effects of context checking, we also examined the following alternatives:

* To island drive or. not: go in both directions from arbitrary starting points in the input versus proceed strictly left to right from the beginning. Island driving allows interpretations to be built up around words that match well anywhere in the input.
* To map all or one: test all the words at

once at a given location versus trying them one at a time and delaying further testing when a good match is found. Mapping all at once identifies the best acoustic candidates and reduces the chances of following false paths, but it takes substantially more time.

* To focus or NOT. assign priorities for tasks focusing on selected alternatives by inhibiting competion or proceed each time with the task with the highest score.

All combinations of the four control-strategy variables were tested on the 60 sentences. This experimental design allows us to compare the 16 combinations of control choices and to evaluate, by analysis of variance the effects of the control strategy variables, that is the change in performance each produces averaged over all the possibilities for the other variables. The probabilities associated with an effect are expressed as p<.01, p< 05, and p<.10, corresponding to the different likelihoods that the effect is the result of random variation.

We will discuss here only the results that show the effects of context checking. The most important performance measures for the system are accuracy (the percentage of sentences for which the correct sequence of words is found) and runtime (the computation required by the system, including simulated acoustic processing time). In the rest of this section, these variables will be considered in order as the results demonstrate the effects of context checking.

First, however it should be noted that the control strategy affects accuracy only indirectly: all the strategies are complete in the sense that they only reorder, and never eliminate, alternatives. If there were no false alarms, all the systems would get 100% of the test sentences correct, Even with false alarms the strategies would get an equal percent correct, if all the possible alternatives could be tried before the system picked an interpretation. Errors would only occur when false alarms had high enough scores to displace hits in the highest rated interpretations. However, in the actual system, the large number of alternatives makes it impossible to consider all of them in the space and time available. As a result, the order in which the alternatives are considered can affect the accuracy, and so can the demands on space and time. Control strategy thus affects accuracy indirectly by reordering alternatives and by modifying space and time requirements. To explain the accuracy effects, we must look at these other factors.

In this experiment, the storage limit had an important influence on accuracy. In the 960 tests (60 sentences times 16 systems) 578 (60.2$) were correct and 382 (39.8%) were wrong. Of the errors, 175 (46%) had an incorrect interpretation, while 207 (54$) had no interpretation at all. Since the systems could potentially get the correct answer, and no time limit was imposed until at least one interpretation had been found all of the 207 sentences with no interpretation were a result of running out of storage.

The storage limit used in the tests was based on the number of phrases constructed When the total reached 500, the system would stop trying new alternatives and if any interpretation had been found pick the highest rated interpretation as its answer The average number of phrases constructed over all systems was 204 nonterminal and 63 terminal. The system with the best accuracy, which included context checking, had the lowest average (113 nonterminals and 45 terminals) while the system with the worst accuracy, without context checking had one of the highest averages (260 nonterminals and 68 terminals). Overall there was a strong negative correlation (- 93) between system accuracy and average number of phrases constructed.

For the 16 system configurations, the values for accuracy ranged from 46.7$ to 73 3$ Averaged over the systems, the inclusion of context checking provided an increase in accuracy of 11.6$, p<.05. Configurations with context checking constructed on the average 240 phrases; those without constructed 294; this difference is significant, p< 01. Surprisingly, context checking also results in a significant reduction in the false terminal percentage -- from 87 5$ to 86.2$ p<.01. This reduction may be evidence that context checking is giving lower priority to looking for words adjacent to false alarms than it gives to looking next to hits. This change could affect the false terminal likelihood, since there is always a hit adjacent to a hit, while false alarms often have nothing but other false alarms next to them. In addition to its effect on false terminals, context checking may also be improving the storage requirements and accuracy by generally improving the priority setting, thereby reducing the likelihood of following false paths.

The important role of the storage limit raises the question of whether the accuracy effects would have disappeared if more storage had been available. We believe that the effects would have been smaller but still important. The effects on the proportion of false terminal phrases would remain, as would the effects on priorities. A smaller percentage of false terminals and better priorities will cause the system to find the correct interpretation sooner and. even if the storage limit were relaxed, the limit on runtime would remain to penalize systems that were slow to find the right answer. The effects of control strategy choices on accuracy would vanish only if space and runtime limitations were both removed.

The system runtime is another important performance measure. Here, we will use the phrase total runtime_ to refer to the simulated acoustic processing, plus the actual processing time (on a DEC PDP KA-10) for the executive and the semantic components. The executive time is mainly spent setting priorities and parsing. The semantics time is used in constructing semantic translations and in dealing with anaphoric references and ellipsis. The reported total runtime does not include acoustic preprocessing or question answering, since neither is affected by the experimental variables. We report results only for total, executive, and acoustic times; semantic times are not reported, both because they are redundant given the other three measures, and because they are relatively small in comparison to the others.

For the 16 system configurations, the values for runtime ranged from 221 to 559 seconds per sentence. Context checking was the only control

variable to reduce total runtime The difference in seconds per sentence of processing time (from 421 to 383) was significant, p<.01. Separating executive and acoustic runtimes from the total again showed that in both cases, context checks decrease the runtime. For executive runtime, the difference is from 117 to 109 seconds p<,10; for acoustic runtime, the difference is from 282 to 254 seconds p< 01.

It is noteworthy that the extra effort for context checking resulted in a net decrease in processing time. For example the best system configuration, which included context checking, spent an average of 6.3 seconds per sentence doing extra processing *for* the context checks; however, this was still 41 seconds per sentence faster than the system configured in the same way but without doing the context checking.

In summary, context checking had uniformly good effects. For both accuracy and runtime, it was worth the extra effort to get better priority setting. This result clearly depends on the fact that we put a large amount of the system's knowledge into the rule procedures of the language definition rather than into the structural declarations. It would be of considerable interest to provide a direct experimental test of the effects of varying the style of the language definition. For example, while keeping the scope of the language constant, a greater amount of the linguistic knowledge could be encoded in the rule structures rather than in the rule procedures. Such an experiment would provide information about the performance effects of the different representational styles and would indicate the extent to which the effects of design features such as context-checking for assigning ratings are dependent on a particular style of language definition. In drawing conclusions from such an experiment, we would be generalizing both over the population of input sentences and over the population of language definition styles.

IX CONCLUSION

The work on speech understanding at SRI has produced system control concepts and a set of system components that are well-suited for further research on speech understanding and also for research on natural language understanding systems with text input. Under ARPA support we are continuing to apply these products of the speech understanding project in a Navy command and control context to provide natural language access to a distributed data base stored on a number of different computers. Under National Science Foundation support. we are exploring the significance of these products for understanding natural language dialog between humans and computers for the accomplishment of a structured task in a dynamic situation. We believe that the elaboration of complex knowledge sources and the development of sophisticated mechanisms for integrating and controlling them will prove to have major implications for future work in both artificial intelligence and computational linguistics.

Contract No. DAAG29-76-C-0012.

Grant No. MCS76-2200H.

REFERENCES

Bernstein, Morton I. Interactive Systems Research Final Report. TM-5243/004 System Development Corporation. Santa Monica California. November 1975.

Fikes. Richard E., and Hendrix Gary G. A Network-Based Knowledge Representation and its Natural Deduction System. Proceedings of *the.* Fifth International Joint Conference on Artificial Intelligence. Cambridge, Massachusetts, 22 25 August 1977.

Grosz. Barbara J. The Representation and Use of Focus in Dialog Understanding. Ph D. Thesis University of California, Berkeley, California, 1977 (a)

Grosz, Barbara J. The Representation and Use of Focus in a System for Understanding Dialogs. Proceedings g£ the Fifth International Joint Conference OJI Artificial Intelligence. Cambridge, Massachusetts, 22 25 August 1977. (b)

Kaplan Ronald M. A Multi-processing Approach to Natural Language. Proceedings National Computer Conference, New York, New York, 4-8 June 1973 AFIPS Press, Montvale, New Jersey, 1973. Pp- 435-440.

Medress, Mark F., et al. Speech Understanding Systems: Report of a Steering Committee. SIGART Newsletter. April 1977, 62, 4-8.

Newell Allen, et al. Speech Understanding Systems. North Holland Publishing Company, Amsterdam, 1973.

Paxton. William H A Framework for Language Understanding In COLING 76, Preprints of the 6th International Conference on Computational Linguistics, Ottawa, Ontario. Canada, 28 June-? July 1976. No. 14. (a)

Paxton, William H. Experiments in Speech Understanding System Control. In Proceedings of the First CSCSI/SCEIO National Conference, Vancouver. British Columbia, Canada, 25-27 August 1976. (b)

Paxton William H. A Framework for Speech Understanding. Ph.D. Dissertation. Stanford University, Stanford, California, 1977.

Reddy, D. Raj. Speech Understanding Systems: Summary of Results of the Five-Year Research Effort. Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania, September 1976.

Walker, Donald E., (Ed.) Speech Understanding Research. Final Report, Project 4762, Artificial Intelligence Center, Stanford Research Institute Menlo Park, California, October 1976.