August, 1973

SPEECH UNDERSTANDING, COMPUTATIONAL LINGUISTICS, AND ARTIFICIAL INTELLIGENCE

By

Donald E. Walker


Artificial Intelligence Center

Technical Note 85
SRI Project 1526

The work reported herein was sponsored by the Advanced Research Projects Agency of the Department of Defense under Contract DAHC04-72-C-0009 with the U.S. Army Research Office.
ABSTRACT

The establishment of a major program of research on speech understanding is likely to make substantial demands on and to have significant consequences for both computational linguistics and artificial intelligence. The ability to talk to a computer in the performance of a shared task requires more than the addition of an acoustic component to a question-answering system. Analytical tools for dealing with semantics and pragmatics are likely to prove particularly important. Furthermore, to the degree that speech understanding requires an integration of linguistic features, it is likely to contribute to a more comprehensive model for language. The current work on speech understanding at Stanford Research Institute is described in the context of these considerations.
INTRODUCTION

The establishment of a major program of research on the development of speech understanding systems is likely to have significant consequences—both for computational linguistics and for artificial intelligence. In a complementary fashion, only through contributions from research on computational linguistics and on artificial intelligence are we likely to be able to talk to a computer in the performance of a shared task. Furthermore, the accomplishment of this goal will require a more comprehensive and more coherent model of language than linguistics now provides. It will be necessary to conjoin acoustics, phonetics, morphology, syntax, semantics, pragmatics, and perhaps other elements of speech and language. To the extent that research on speech understanding is successful, the systems developed may help linguists improve their models.

Computational linguistic analyses of the structure of language have been concerned primarily with syntax. It is possible to characterize sentences as strings of words, because the form of the textual input provides that preliminary "parsing." The words themselves can be identified using limited or expanded dictionaries and various strategies for morphological analysis. The surface and deep structures of these strings then can be determined by algorithms that involve reference to sets of grammatical rules. Parsing systems for syntactic analysis have had to cope with the effects of typographical errors, words that lack dictionary entries, and sentence patterns that are not well formed with respect to the grammar. However, the presence of a keyboard device constitutes a buffer that allows the user to see the form of the input and, in response perhaps to system probes, to edit and modify it selectively in the direction of acceptability. Talking to the computer through a microphone changes the situation dramatically.

Utterances spoken in a conversational manner are not segmented into
words in any consistent fashion. Occasionally, pauses occur in the
proper places, and sometimes prosodic features like stress and intonation
provide information on boundaries, but the results are not reliable and
certainly are not well understood. In addition, a speaker's pronunciations
of a word vary: in relation to the phonetic context of the preceding and
following words; in relation to stress, pitch, and location relative to
the overall intonation contour of the utterance; and in ways that seem
arbitrary and capricious—at least, we do not understand them. Moreover,
people often do not speak in complete, syntactically well-formed sentences.
There are errors and uncertainties associated with the production of an
utterance and with its transmission and reception as well. Also, the
conventions for conversations allow a speaker to use pronouns and other
words whose referents depend on previous utterances (by himself or by
others)—anaphora; spatial or temporal designations may be implied—deixis;
and it may be appropriate to omit words or even entire sentences—ellipsis.
In view of these factors, it is clear that understanding speech will require
new advances in computational linguistics.

There has been a substantial amount of research directed toward the
recognition of speech (see Hill, 1971, and Lea, 1972), but it has never been
considered a proper concern of computational linguistics. The work focuses
primarily on acoustic features of the speech signal, and procedures involving
pattern matching frequently are used. The systems developed so far make it
possible to identify isolated words selected from relatively small vocabularies
and spoken by a few people carefully chosen because of their voice character-
istics, or for whom the acoustic parameters have been adjusted. Some commercial
versions of such systems are available. However, attempts to extend word
recognition approaches so that they handle continuous speech, have not proved
successful. It has long been acknowledged that more linguistic information
is required, but it also seems likely that other procedures will be necessary
to constrain the number of ways that portions of the acoustic wave form for an
utterance might be interpreted. It is in this context that research on artificial intelligence becomes relevant.

People working on artificial intelligence have been concerned (among other things) with algorithms and heuristic procedures for problem solving and for the performance of complex tasks. Reduction of the search space—the number of alternative possibilities that need to be considered—often is a critical requirement. Until recently, the orientation has been toward abstract characteristics of that space, resulting in techniques for optimal path finding, graph traversing, or tree pruning from arbitrary structures. Now, the major concern is how to represent the semantics of particular task domains. In the area of vision, for example, one of the primary factors in the identification of an object in a scene is expected to be its location in relation to other objects with which it is associated naturally. Thus, a chair is frequently found in the neighborhood of tables and desks. For more complex problem solving, and particularly for research on robots, it is essential to complement semantics with knowledge about the pragmatics of actions in the task domain. That is, information must be provided about the goals of behavior in a given situation and about progress toward these goals. These, and probably other aspects of research on artificial intelligence, will prove relevant for the development of speech understanding systems.

AN INTEGRATED SPEECH UNDERSTANDING SYSTEM

The approach to speech understanding at Stanford Research Institute combines computational linguistic techniques with strategies developed in research on artificial intelligence and with procedures for acoustical analysis. The goal of the research is a system capable of engaging a human operator in a natural conversation involving a specific task domain. In recognition of the complex factors involved in this effort, we considered it of primary importance to establish a functioning system as soon as possible. While
many of the problems can be anticipated in advance, it is clear that the most critical ones arise in the integration of the various sources of knowledge that are required, and little relevant research is available as a guide. As a result, for our first system implementation we began with Winograd's programs for natural language understanding (Winograd, 1971).

The task domain used by Winograd involved the simulation of the actions of a robot that knows about and can manipulate blocks of various shapes, sizes, and colors. Our intention was to develop a system that would allow a person speaking into a microphone to ask questions about this "blocks world," to give commands that would modify it, and to add information that would augment its structure. During the first year of the project, we completed a system, based on Winograd's work, that brought together syntactic, acoustic, and semantic information in the analysis of spoken utterances. The results, although far from satisfactory for actual speech understanding, provided valuable guidance for a major modification of the system implementation. Furthermore, the results encouraged us to continue working with the particular design concepts we had developed.

During this second year of the project, we have completed a second version of our speech understanding system, one that we believe can be refined and improved to provide the capabilities needed to achieve our goal. For this version, we have established a new task domain, the assembly and repair of small devices like faucets and pumps. Our intention is to allow a person to give directions to guide the system, to ask questions about available tools or parts, and to add information so that, for example, particular characteristics of a given faucet could be related to other faucets the system has had experience with in the past.

The emphasis on task domains, in the description so far, is deliberate. In contrast to the goals of research on speech recognition, the goals of speech understanding research are both broader and narrower. Speech recognition
efforts have been directed toward providing an orthographic transcription of the sounds and words corresponding to the acoustic signals of arbitrary utterances. Although the vocabulary might be limited, the intent is to process arbitrary utterances, essentially independent of context. Speech understanding is broader because it seeks to interpret the meaning of an utterance rather than just identify its form. It is narrower, because it can expect to process only those utterances that are relevant to a particular task domain. Our commitment to a speech understanding approach is based on a conviction that a speech recognition system, as described above, will not be successful with continuous speech. At the same time, we appreciate the complications that result from having to understand an utterance so that an appropriate and meaningful response can be made to it.

The System Design

The design concept underlying the Stanford Research Institute speech understanding system is to use knowledge of various kinds to predict the sequence of words that has been spoken. Our efforts have been concentrated on syntactic, semantic, and pragmatic information, but we know that it will be necessary to incorporate prosodic data and to use acoustic evidence to focus the predictions. However, we want to minimize the demands on acoustic processing; in particular, we believe that hypothesizing words—and even phonemes—solely on the basis of acoustic data is not only costly but also adds undesirable complications. Words in an ordinary conversation vary so much in pronunciation, that a refined segmentation and labelling can be too specific. The use of confusion matrices to indicate similarities in sounds is helpful, but it does not take context sufficiently into consideration. The mechanism we are using is to constrain the search space, so that at a particular place in the utterance the system will be testing for a specific word. An acoustic characterization of the word is compared with the acoustic data preprocessed for the utterance, essentially as a procedure or program—not as a pattern matching exercise. These word functions provide a basis for weighting more heavily the distinctive and
distinguishing acoustic features of the word in that context, and for minimizing the others.

In order to use a variety of sources of knowledge effectively, it is necessary to have some way of coordinating them. We are using the parser for this purpose; successive choices through the parse tree are influenced by information from the grammar, from the task domain, from the current state of the analysis, and from as many other sources as we can formalize.

System #1

As stated earlier, the first version of the Stanford Research Institute speech understanding system made use of Winograd's "blocks world" task domain and his procedures for analyzing natural language. As used by Winograd, sentences were typed in. A person might direct his system to 'Pick up a red block.' 'Put it on the blue block.' 'Build a steeple.' The system might respond, 'I don't know the word "steeple".' The user could define it as 'Two cubes with a pyramid on top.' and continue. A large variety of constructions could be understood, with complex anaphoric references.

For speech understanding, we were able to take Winograd's grammar, essentially as described in his thesis (Winograd, 1971), with only minor modifications. We also were able to make use of the semantics he embodied in MICROPLANNER code. However, major changes in the operation of his parser were required. PROGRAMMAR depended on feature information provided by lexical lookup of the successive words in the typed input string. Since we needed to use the parsing procedure to help segment and identify the words in speech input, it was necessary to find other ways to control the generation of paths through the grammar. Consequently, we changed the parser so that syntactic and semantic constraints could be used to influence successive choices through the parse tree. It also proved necessary to add an interpreter to provide a backtracking mechanism. (Winograd had tested and eliminated least likely alternatives first and thus did not require one.) Details on these
modifications and on other aspects of this first system can be found in two other papers (Walker, 1973a, 1973b).

The syntactic and semantic information made it possible to predict words likely to be present at a particular place in an utterance. These predictions were tested by comparing word functions—that is, acoustic characterizations of the words—against the acoustic data preprocessed for the utterance. After the speech input was recorded, the analog data were digitized and then processed in two separate ways. A digital filter analysis provided the basis for classifying each 10-millisecond segment into one of six crude, but highly reliable, categories: silence, unvoiced turbulence, voiced stop, vowel-like, and other. A linear predictive coding (LPC) analysis, using an algorithm developed by Markel (1971), provided formant frequency and amplitude data. The results of these analyses together with the segment classification were stored in disk files.

Each word function was prepared on the basis of a careful examination of the acoustic characteristics of the word as recorded in a variety of contexts. Beginning at a place in the acoustic stream designated as the approximate location of the start of the word, a search of the segment classifications was conducted to locate a vowel-like string of sufficient duration. Vowels most often prove to be the most stable element in a word. The formant frequencies of this string were calculated to determine whether they fit into a range considered appropriate for the vowel in the given word. Then the segments preceding the vowel-like string were examined to establish their correspondence to the consonant or consonant cluster (if any) in the word based on the segment classifications. Similar comparisons were made with segments following the vowel-like string. Confidence ratings were given for the results of each comparison (positive, possible, unlikely, impossible), and a composite rating was returned for the whole word.

In testing the procedures described in this section, we processed sentences using word functions for only some of the words. For example,
'Put the black block in the box.' was analyzed correctly with a typed-in context of 'put the' and 'in the'. On the basis of our experience in developing this first version of our system, and influenced by the results of a minimal program of testing, we began the major revision that led to System #2.

**System #2**

The use of Winograd's programs made it possible for us to complete a system and to get some information about the reasonableness of our system design much earlier than we would otherwise have been able to do. However, it was obvious, almost from the beginning of our work, that the "blocks world" was not an appropriate task domain. In order to take advantage of the cumulative semantic and pragmatic constraints that could in principle be provided by our system design, we needed a domain that would establish a relatively complex task that would be completed through a conversation involving the achievement of a number of subgoals. The domain chosen, as we have already indicated, is the assembly and repair of small mechanical devices. We are beginning with an ordinary sink faucet. A person using the system will guide the system in identifying the source of a leak in the faucet and will tell the system how to repair it.

With regard to bases for predicting words, our experiences with the first system made two things clear. First, it would be necessary to develop and add more complex and sophisticated sources of knowledge. Of course, we had anticipated that requirement. Second, we came to realize that the variety of sources of knowledge required—including user models, prosodic data, and other acoustic data as well—made it necessary to make a major change in our parsing strategy. As a result, we have developed a "best-first" parser that contrasts with both depth-first and breadth-first approaches. It is described in detail in a paper by Paxton and Robinson (1973), and will only be summarized briefly here. The parser is designed to deal flexibly with decision making at choice points in the grammar,
particularly where there are several alternatives for continuing the parse, yet, not enough information to select one among them. In contrast, a depth-first strategy, like Winograd's, requires pursuing a single path from a choice point until it is demonstrated to be inconsistent with the input; then backtracking is required to return to that position. The results are very uneconomical; moreover, in parsing speech, the uncertainty of the acoustic data makes it undesirable to decide conclusively that a particular path should be abandoned. It would seem more appropriate to consider the likelihood of a path rather than its acceptance or rejection, and it would be preferable not to have to exhaust all the alternatives following from a particular choice point before abandoning it.

The best-first design is based on heuristic search strategies developed in research on artificial intelligence. As Paxton and Robinson describe it, "In this approach, each new path resulting from a choice point is assigned a priority according to its estimated likelihood of leading to a correct parse. The paths are then added to the set of all paths that have been generated but not yet extended during this parse. The system follows the highest priority path until its priority drops or it reaches a choice point. At that time the cycle repeats. Since the new path chosen need not be one of the successors of the previous path, the parser will not necessarily continue along a single path until it reaches a dead end. Instead, it will suspend a path when there is an alternative available with a higher estimated likelihood. It will resume the original at a later time if it becomes most likely again." (p. 217)

An initial implementation of a best-first parser has been completed. It includes an interpreter for LISP extended to allow multiprocessing control structures like those described by Bobrow and Wegbreit (1972). Eventually, LISP itself will provide such control structures, and programs written for our interpreter will run directly in LISP. We have made considerable progress on the problem of effectively sharing information among the competing processes.
looking for a parse. Our initial attempts involved reusing the results of acoustic tests for words proposed in the input stream and saving maps of successful parses of various constituents (see Paxton and Robinson, 1973, for details). More recently, we have used the multiprocessing control structures to implement a scheme in which a single family of processes acts as the sole producer of certain constructions, e.g., noun phrases, beginning at a certain location in the input. Processes needing to find such constituents both provide a range of contexts to be considered when establishing priorities within the family and act as consumers of structures produced by the family (see Kaplan, 1973, for description of a similar procedure that has influenced our work).

The current grammar for System #2 contains clause, noun group, and verb group subgrammars, modified extensively from those used in the first system. A case grammar has been added to reflect the possible surface arrangements of verb case arguments in an utterance. The paths of the clause subgrammar establish the following kinds of clauses: major (declarative, imperative, or question), adjunct, sentence complements, nominalizations, and noun qualifiers. The verb subgrammar establishes verb groups that are present, past, or future tense; active or passive voice; simple, emphatic, or progressive form; and perfect or imperfect. An auxiliary may be predicted preceding and separate from the main verb group in a question. Determiners, adjectives, quantifiers, pronouns, nouns, and qualifying phrases and clauses are established according to paths in the noun group subgrammar.

The case grammar, influenced in particular by the work of Celce-Murcia (1972), provides a basis for incorporating semantic information. The constraints set by the particular sense of the verb being processed predict the presence of certain constituents: particles, noun groups, preposition groups, complements. This information is used to set priorities in the parser. It also provides a mechanism to relate knowledge about the current state of the task domain to decisions about the choice of paths.

A simple model of the "faucet world" has been built, using QLISP, an extension of LISP to include complex procedures. (QLISP is being developed
by Reboh and Sacerdoti (1973) at SRI based on earlier work of Rulifson, Derksen, and Waldinger (1972) on QA4.) The model now contains a number of objects related to faucets and their repair, each with various relevant properties. Various relations exist among these objects—all spatial at present, and several simple actions can be performed. Whenever an action takes place, the model is updated accordingly. A "memory" allows questions to be asked about previous actions, but not yet about past states of the model; it also provides a basis for identifying anaphoric references. Information from the "faucet world" is being used to affect certain priorities in the parser. After a noun phrase has been processed, the presence of a definite article or other quantifier results in a procedure call to check on its correspondence with the current configuration of that world.

The procedures for acoustic preprocessing of an utterance have been refined and extended in a number of ways. Additional digital filters have made it possible to subdivide turbulent sounds into six classes: s, sh, f/th, and their voiced counterparts. Thus, each 10-millisecond segment is now classified reliably as one of ten categories. Spectral data from the LPC analysis provide a segmentation of voiced and vowel-like sounds on the basis of algorithms that establish formant trajectories and identify formant amplitude levels. The word functions that we have written make use of the added acoustic information. Although still relatively few in number, the dozen or so already developed have made no errors for stressed words when processed against a fairly large body of utterances.

System #2 is now in operation; the parser has been linked to the acoustic routines, and a few sentences have been processed. The analyses are correct, but we recognize that these preliminary results do not constitute a test of the system.

Some remarks about implementation may be in order, even though it is clearly in an early stage of its development. It is running on the SRI PDP-10 computer under the TENEX operation system. The LISP code for the
parser and the multiprocessing interpreter occupy about 64K words of core, 
and the QLISP package adds about 52K more. The FORTRAN load package for the 
acoustic routines takes about 100K words, including 50K of data for an 
average utterance. A substantial amount of the program size is devoted to 
debugging and development aids. Eliminating those aids, optimizing the 
code, and using special purpose hardware would make the system more efficient, 
but such actions are not called for at this time. In any case, it is clear 
that an effective system able to coordinate all the information necessary 
for speech understanding will be large regardless of how it is packaged. 
Also, the system must be able to run approximately in real time—but we are 
deferring consideration of that objective for a while.

Future System Development

More work needs to be done before we can evaluate our approach 
to speech understanding. We have a parser designed to handle multiple 
Sources of knowledge, but we need to supply that information to the parser 
just where and when it can be used. The sources of knowledge themselves 
need expansion, elaboration, and even creation. We have a preliminary 
world model and the bare beginnings of a task model and a dialog model. 
The task model must provide guidance through the subgoals involved in 
faucet repair. The dialog model must allow each interaction to be related 
to all relevant preceding interactions between the user and the system; it 
also may be helpful in predicting some of those to follow. The user model 
is clearly an exercise for the future; characteristics of particular individuals may affect their choices of constructions in the grammar, words in the 
lexicon, or steps in the solution. However, first we need procedures that 
work for some "standard" case.

We know that prosodic information will be essential, and we are 
planning to devote a substantial amount of effort to the development of 
procedures that will allow us to use it. Prosodic data perform some of
the functions for spoken language that punctuation does for text. Intona-
tion contours might help predict sentence type; together with stress and
pause, they should help identify clause and phrase boundaries and point to
parts of an utterance with special semantic import (see O'Malley, 1973).
Other acoustic data, derived from the particular portion of the utterance
being considered at the moment, could be used to order the testing of words.

Refinements of the grammar can continue indefinitely, of course.
The present grammar appears to be as extensive as others we are familiar
with, in particular those of Winograd (1971) and Woods et al. (1972),
against which it has been directly compared. Of particular importance to us,
because of the role played by the grammar in the operation of the parser, will
be ways of incorporating semantic information more directly. Future grammars
for spoken English will be required to accommodate departures from "grammatic-
cality." Our expectation is that the result will be systems of rules in
which syntactic and semantic information are blended.

Our continuing work on acoustics will be directed toward more refined
classifications of ambiguous elements. In particular, we need to disambiguate
stops and nasals and to classify liquids and glides. Our current word
functions handle intra-word co-articulation—the effects of immediate context
on sounds within a word—quite well, for those words we have done. We need
to improve them, to prepare a more general framework for writing them, and
to extend them to handle co-articulation between words. We have developed
our present word functions using only two speakers; their speech differences
have not made it necessary to modify any algorithms, and changes in thresholds were
required only for fricatives. However, more work needs to be done on speaker
variation, before we can be comfortable that the word function strategy can
handle it adequately.

Our plans for future system developments provide us with many things
to do. It is likely, though, that additional requirements will emerge in
the course of our work. We are two years into what began as a five-year
project, and we are working in close coordination with eight other contractors
in the ARPA Speech Understanding Research Program (see Newell, et al., 1973, for information about the problems and the plans that led to the establishment of this program). Nevertheless, although we are enthusiastic about the current state of the work and about the likelihood of our success, it is clear that the results presented here are only a beginning.

CONCLUSION

Only a few remarks seem appropriate in conclusion, since so much is left for us to do. We began by discussing the relations between speech understanding, computational linguistics, and artificial intelligence, and this paper, in essence, testifies to their interdependence. Research on speech understanding can benefit from and can motivate research in the other areas; both consequences are desirable. Whether the modeling of language needed for effective speech understanding will be helpful to linguistics remains to be determined. We believe that we will need a level of sophistication for our model that linguists would find respectable. We hope they can become interested enough in our work to provide some guidance along the way.

ACKNOWLEDGEMENTS

A large number of people at Stanford Research Institute have contributed to this project during its two-year period. The most recent work has been sustained primarily by Sharon Baranoisky, Dick Becker, Earl Craighill, Grant Hoyt, Bill Paxton, Tito Poza, Ann Robinson, and Jim Young. Some of their words have even crept into this paper. We are also indebted to other participants in the ARPA Program on Speech Understanding Research.
REFERENCES


O'Malley, Michael H., "The Use of Prosodic Units in Syntactic Decoding," Phonetics Laboratory, University of Michigan, Ann Arbor, Michigan (1973).


