American Journal of Computational Linguistics *Microfiche 34 : 5*

SYSTEM INTEGRATION AND CONTROL IN A SPEECH UNDERSTANDING SYSTEM

WILLIAM H. PAXTON AND ANN E ROBINSON

Artificial Intelligence Center
Stanford Research Institute
Menlo Park, California 94025

ABSTRACT

Two important problems in speech understanding are how to effectively integrate multiple sources of knowledge within the system and how to control the activities of the system to arrive at appropriate interpretations for utterances. This paper first describes the roles played by acoustics, syntax, semantics, and discourse, and shows how a language definition is used to integrate them into a system in a way that allows the interactions to be easily visible. The second part of the paper describes an executive that uses information from these knowledge sources in its control strategy.

Acknowledgment

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and monitored by the U.S. Army Research Office under Contract No. DAHC04-75-C- 006.

A speech understanding system must use many kinds of knowledge, each playing a particular role during the interpretation of an utterance. While these roles are interrelated, it is important to be able to separate the knowledge sources so that interrelations are visible and so that the contributions from the various sources can be studied. The knowledge sources used in the system being developed jointly by SRI and SDC can be characterized broadly under the headings of acoustics, syntax, semantics, and discourse (Walker et al., 1975; Robinson, 1975; Hendrix, 1975; Deutsch, 1975; Slocum, 1975; Pitea, 1975).

The acoustic component relates linguistic entities (words and phrases) to the speech waveform. An acoustic-phonetic processor analyzes the digitized waveform to extract parameters based on speech production characteristics. The parameters include fundamental frequency, voicing label, formant frequency, energy data, and others. Following parameterization, various rules are applied to generate an acoustic feature description of the utterance. The parameters and features are subsequently used by the lexical mapping procedure. The mapper is called during the parsing of an utterance to give a decision score as to whether a proposed word or phrase could actually be present in a specified time region of the input. Phonological and acoustic-phonetic rules are used by the mapper to relate phonetic spellings to acoustic data.

Syntax provides reliable, reasonably inexpensive indications of which words or groups of words may combine and of how well they fit. Syntactic rules give general patterns for constructing noun phrases, clauses, and sentences and provide consistency checks for such items as number agreement. In testing word or phrase combinations, syntactic information alone can often rule out a candidate without the need for more costly semantic and discourse analysis.

The semantic component includes a general model of- the domain of discourse, and a set of algorithms for combining (or rejecting) concepts in the domain. For example, given a verb and two noun phrases, semantic routines can build the corresponding semantic relation between the items indicated by the noun phrases.

The discourse component deals with the relationship of the current utterance (or a portion of it) to the dialog context and to entities in the task domain. Discourse functions use information from previous utterances to fill out elliptical expressions and to find referents for pronouns and definite noun phrases.

The language definition is the focal point for integrating these knowledge sources. A language definition includes (1) sets of units out of which utterances in the language are constructed and (2) rules for combining the units into larger structures. The basic units will be called 'words' (although this technical use does not exactly correspond to the common use). The composition rules indicate how phrases can be combined into still larger phrases. More precisely, a, phrase' is either a word in the input or the result of applying a composition rule to constituent phrases. The rules give the linear pattern of constituents and specifications for calculating values for both the attributes of the resulting phrase and for factors used in judging the result.

It is at the phrase level that the knowledge sources are integrated into the system. There are two aspects to the contributions from each source: the values of properties of the phrase as computed by the knowledge source, and the source's assessment of the correctness of this phrase as an interpretation of the input. These two aspects are reflected in the attribute and factor statements that are associated with each of the words and phrases in the language definition. The attribute statements provide instructions for computing various properties of the phrase. These instructions may call upon any or all of the sources of knowledge. For example for a phrase spanning a particular segment, an acoustic attribute may specify the words in that segment; an attribute supplied by the syntax can specify a feature such as the voice ('active' or 'passive'); an attribute supplied by semantics can specify a semantic net interpretation built from the semantics of the constituents; and an attribute supplied by the discourse component can indicate a referent or an implied meaning.

Factor statements tell how to use these attributes in determining the likelihood that the phrase is a correct interpretation of the input. The result of combining the factors for a particular phrase is called a score. The use of such scores by the executive in determining overall strategy is described below. Factors are nonbinary; since they can have a range of values, rigid 'yes' or 'no' decisions do not have to be made in assessing the quality of a phrase. For example, the closeness of the acoustic match may vary and this can be reflected in the corresponding factor. Weak evidence from one source of knowledge could lower the score, while strong evidence from another source could compensate for that and actually raise the score.

In summary, a phrase is a composite interpretation of a particular portion of the utterance, integrating contributions from all relevant knowledge sources. This means that each portion of the input is interpreted and evaluated by the system as fully as possible, as soon as possible. The system is never faced with the problem of relating or combining fragmentary theories constructed independently by different knowledge sources, and evaluations made by different sources are immediately merged to control and coordinate overall system activity. For example, as soon as a definite noun phrase is found, the acoustic component checks the coarticulation of the constituents, the syntactic component checks for agreement in features such as number, the semantic component builds a representation of the meaning, and the discourse component looks for a referent.

The following example illustrates how several knowledge sources are used together to interpret and evaluate phrases. The rule shown is for the composition of a noun phrase such as 'what submarine' or 'their submarines' and illustrates the integration of acoustic, syntactic, semantic, and discourse information.

```
RULE.DEF NP7 NP = DET NOM;

  ATTRIBUTES

    STRING = APPEND(STRING(DET),STRING(NOM)),
    NBR = GINTERSECT(NBR(DET),NBR(NOM)),
    CMU = GINTERSECT(CMU(DET),CMU(NOM)),
    SEMANTICS = SEMCALL("SEMRNP7,SEMANTICS(NOM),
      MOOD(DET),GCASE(DET),INTERPRETATION(DET)),
    DISCOURSE = IF MOOD(DET) EQ "DEC THEN
      DISCALL("DISRNP7,SEMANTICS) ELSE "UNDEFINED,
    INTERPRETATION = IF DISCOURSE NQ "UNDEFINED THEN
      DISCOURSE ELSE SEMANTICS;

  FACTORS

    COART = MAPPER(LASTWORD(DET),FIRSTWORD(NOM)),
    NBR = IF NULL(NBR) THEN OUT ELSE OK,
    CMU = IF NULL(CMU) THEN OUT ELSE OK,
    SEMANTICS = IF NULL(SEMANTICS) THEN OUT ELSE OK,
    DISCOURSE = IF MOOD(DET) NQ "DEC THEN OK ELSE
      IF NULL(DISCOURSE) THEN POOR ELSE
      IF AMBIGUOUS(DISCOURSE) THEN OK ELSE GOOD;

END;
```

The first attribute statement computes the STRING of the resultant phrase, which is an acoustic attribute indicating the words composimg this phrase. NBR (number) and CMU (count-mass-unit) are syntactic attributes for the phrase, each being derived from the intersection of the corresponding attributes of the constituents. The semantics attribute is a piece of semantic net that is constructed from the semantics of the constituents by the semantic routine (SEMRNP7) associated with this rule. If the MOOD attribute of the DET constituent is *DEC, i.e., a declarative determiner, then the discourse routines will look for a referent for the phrase in the dialog context and assign its semantic structure as the value of the attribute DISCOURSE. The INTERPRETATION of the phrase is either the referent found by discourse or the semantic net structure in case no direct reference is found.

The factor statements use these attributes in computing contributions towards the score for the phrase. As has been mentioned, there is a range of acceptable values for factors. For simplicity, symbolic values are used (VERYGOOD, GOOD, OK, POOR, BAD, and OUT). In the example rule, there are factors determined by each of the major knowledge sources. The COART factor reflects an acoustic test of the coarticulation of the last word of the determiner and the first word of the nominal. NBR and CMU are syntactic factors that will eliminate the phrase if either attribute is incompatible between the constituents. The semantic factor will eliminate the phrase if no semantic interpretation can be formulated. While the current semantic component does not have a metric for determining the likelihood of an interpretation other than whether or not a semantic representation can be built, it is possible to introduce such a metric and have the semantic factors be nonbinary. The discourse factor is nonbinary. If the determiner is declarative, the discourse has tried to find a referent. If no referent was found, the factor is given a low value, 'POOR', but the phrase is not discarded. If several possible referents were found, the phrase is kept and the score is not lowered because the ambiguity can perhaps be resolved later. If just one referent was found, it is taken as evidence that the phrase is a correct interpretation for that portion of the utterance and the factor is given a higher value 'GOOD'.

The example discussed above shows how the language definition system can be used to integrate a variety of knowledge sources in a way that keeps the contributions and interactions of the different sources easily visible. The representation combines procedural information (in the expressions for calculating attribute and factor values) and declarative information (in the constituent pattern) in a form designed to simplify the task of writing a large definition containing many rules. However, before the rules can actually be used, they must be converted to a different representation designed with efficiency in mind. This translation is done by a language definition compiler' that constructs an internal representation of the language definition that depends in an intricate way on the structure of the 'executive', the portion of the system responsible for scheduling and controlling the various tasks to be performed in constructing an interpretation of an utterance. The operation of the executive is the subject of the rest of this paper.

The executive makes a distinction between the phrases being built and the tasks required to build these phrases. A data structure, called the 'parse net', represents the growing collection of phrases, and another structure, called the 'task queue', encodes the alternative operations available for taking another step toward understanding the input. Each entrv in the task queue specifies a procedure to be performed at a particular location (node) in the parse net. The performance of such a procedure typically entails both modifying the parse net and scheduling new tasks to make further modifications. Each task has associated with it a priority for performing it. The method for determining priorities is described below.

Tasks can include looking for a new word or phrase to finish an incomplete phrase (one missing some of its constituents) and trying to use a word or completed phrase in a larger phrase. This means that the system can work both 'top down' and 'bottom up', because it can look in a goal-driven manner for missing constituents of higher level phrases, and it also can accept words from the acoustics to build into larger phrases in a data-driven manner. As an example, consider the simple grammar with the following composition patterns:

```
S = NP VP
VP = VP NP | VERB
VERB = own | lost
NP = they | the house | it
```

Assume that the word 'they' has been found initially either by the acoustics directly or as a result of confirming a prediction made by the language definition, 'They' constitutes a complete NP, This NP can be put into the S rule, causing the partially filled phrase 'they VP' to be added to the parse net, Already, some of the attributes and factors for the S rule can be determined, and a score computed for this phrase, Building this partial phrase leads to the creation of a new task: to look for a VP following the NP, That task in turn leads to two alternative subtasks: look for a VP NP or look for a VERB, Priorities for both these tasks are computed and they are put on the task queue to be processed, The executive then removes the next task from the queue and continues,

In general, deciding which task to perform is of great importance, because only a subset of the scheduled tasks will actually prove to be necessary to understand the input; the others will be 'false steps' leading to dead ends, Ideally, in deciding which task to do, the executive would always choose one of the necessary tasks and never take a false step, The utterance would be understood with the unnecessary tasks still left in the queue, To approach this ideal, the actual system must spend some of its effort in choosing tasks, Such effort is well spent if it produces a net decrease in processing time, In other words, the efficiency of the system will be improved by decisions regarding the order in which tasks are performed, if the cost of the decisions is less than the cost of the false steps that would otherwise have been taken. Since acoustic uncertainty in speech understanding makes the potential for wasting effort on unnecessary operations particularly large, the system can afford to carry out rather complex computations in deciding what to do next and still obtain a large improvement in overall efficiency. In the current system, the decisions are based on the relative priorities assigned to the various tasks waiting in the queue. Tasks are associated with phrases, and task priorities largely depend on how important the system feels it is to process the phrase.

In addition to the scores of phrases, which combine a variety of factors but are independent of the larger sentential context, the system forms another assessment of the quality of the phrase called the phrase 'value', which depends on the context of proposed complete interpretations for the entire utterance. The phrase value is an estimate of the highest score for all possible interpretations spanning the utterance that include the phrase. The estimate is computed by means of a heuristic search of the space of possible sentential contexts established during the previous tasks performed by the executive.

The priority of a task is initially set to the value of its associated phrase, but the priority is lowered if the task conflicts with the executive's current 'focus of activity'. The phrase value that determines the initial priority reflects an evaluation of both the internal structure of the phrase and its relation to its context, but it does not reflect its competition. If a phrase has a high value, other similar phrases are also likely to have high values. If values alone determined priorities, then even after successfully completing a phrase, the system would tend to continue looking for minor variations in the same area rather than moving on to look for ways to construct a complete interpretation. The 'focus of activity' mechanism provides a way for phrases to inhibit the executive from looking for competing phrases that would necessarily replace them. This focusing is brought about by lowering the priority of tasks that look for replacements for any of a set of focus phrases, until the potential replacement promises to lead to a significant improvement in value for the final interpretation. The effect is to bias the executive toward building up a complete interpretation using phrases in focus rather than exploring competing interpretations that would not use focus phrases. If the focus is wrong, the attempts to extend it to a complete interpretation will be unsuccessful. Eventually a task that conflicts with the focus will become the highest priority operation for the executive to perform in spite of the bias against it. As a result the focus will be modified so that it is consistent with the new task, and the executive will then concentrate on using the revised set of phrases.

In addition to calculating priorities of tasks on the basis of phrase values and focus of activity, the executive must ensure that the information gained through the performance of the tasks is used effectively. This is done by structuring the parse net and the tasks that operate on it in a way that brings together related activities and coordinates them to eliminate duplication of effort. By avoiding duplication, the system reduces the ill effects of the false steps it will inevitably take. Work done on a false path is not necessarily wasted, since it may produce a phrase that can be used in some other way. For example, a phrase constructed as part of an unsuccessful search for one type of sentence may later appear in the final interpretation as part of a different kind of sentence. Also, false steps are not repeated, since the system only makes one attempt to build a particular type of phrase in a particular location in the utterance, regardless of how many larger phrases might include it. Mistakes are inevitable, but at least the system will not make the same mistake twice in one parse

To summarize, the language definition is designed to facilitate the integration of many knowledge sources. Rules in the language definition contain attributes and factors from all of these sources. The attributes are used to indicate particular properties of phrases, and factors then use these attributes to determine the score of the phrase. The external representation of the language, designed for easy use by people, is converted by a language definition compiler into an internal representation, designed for efficient use by the executive. In a step by step manner, the executive uses this information to create, evaluate, and combine phrases. The choice of the next operation to carry out takes the form of assigning priorities to alternative tasks. Priorities reflect both the expected values of complete interpretations toward which the task would lead and the relation of the task to the current focus of activity. Finally, the entire process is organized so that information gained in performing a task is shared and recorded in such a way that it does not have to be rediscovered.

References

Deutsch, Barbara G. Establishing Context in Task-Oriented Dialogs. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975.

Hendrix, Gary G. Semantic Processing for Speech Understanding. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975.

Ritea, H. Barry. Automatic Speech Understanding Systems. Proceedings, Eleventh Annual IEEE Computer Society Conference, Washington, D.C., 9-11 September 1975.

Robinson, Jane J. A Tuneable Performance Grammar. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975.

Slocum, Jonathan. Speech Generation from Semantic Nets. Presented at the Thirteenth Annual Meeting of the Association for Computational Linguistics, Boston, Massachusetts, 30 October - 1 November 1975.

Walker, Donald E., et al. Speech Understanding Research. Annual Report, Project 3804, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California, June 1975.