CONTINUOUS SPEECH UNDERSTANDING

W.A. Woods

Rapporteur: **Dr. J.M. Rushby**

Abstract

The talk describes the research and development in speech understanding systems carried out at Bolt Beranek and Newman Inc. (BBN) during the ARPA Speech Understanding Project. This work included the development of bottom-up and top-down acoustic-phonetic recognizers, a lexical matching scheme that accounts for within-word and across-word phonological effects, for the use of ATN grammars, a uniform scoring philosophy for combining the evaluations of different knowledge sources, and the exploration of approximate and admissible control strategies. These developments were implemented in a speech understanding system called HWIM (for "Hear What I Mean").

1. Characteristics of the Speech Understanding Process

A naive view of speech understanding might consider it as a process of successively recognizing speech sounds (called phonemes), grouping phonemes into words, parsing word sequences into sentences, and finally interpreting the meanings of those sentences. However, considerable experience now indicates that the acoustic evidence present in the original speech signal is not sufficient to support such a process (Woods and Makhoul 1974). For sentences recorded from continuous speech, it is not generally possible to reliably determine the phonetic identity of the individual phonemes (or even to be sure how many phonemes are present) using the acustic evidence alone. Experiments in spectrogram reading (Klatt and Stevens 1971) indicate that the reliability of such determinations can be increased by use of the redundancy provided by knowledge of the vocabulary, the syntax of the language, and semantic and pragmatic considerations.

Tape splicing experiments (Wanner 1973) seem to indicate that this low-level acoustic ambiguity is an inherent characteristic of continuous speech and not just a limitation of human spectrogram-reading. Specifically, intelligibility of individual words excised from continuous speech is very low, but the intelligibility increases when sequences of two or three words are used. It appears that the additional constraint of having to make sense in a larger context begins to resolve the ambiguities that were present when only the acoustic evidence was considered. This processing, however, happens below the level of introspection and has all the subjective characteristics of a holistic or Gestalt phenomenon. That is, if a sufficiently long sequence of continuous speech is heard, its correct interpretations usually appears immediately and effortlessly, without conscious awareness of the details of the process. The vast majority of our spoken communications are understood in this manner, and it is markedly conrasted with those cases where an utterance is garbled sufficiently to invoke conscious effort to decide what was said.

Recently, speech understanding research has taken a direction that recognizes the importance of syntactic and semantic constraints as an essential part of the process which deciphers speech signals into sequences of sounds (see Newell et al. 1973). Consequently, it has become important for speech researchers to be acquainted with the work that has been done in the area of computational linguistics, attempting to construct computer programs to model the process of natural language understanding. In this paper, I will attempt to provide an introduction to some of the ways that these "higher level" sources of knowledge can be used in speech understanding.

2. Syntactic and Semantic Analysis

There are two parts of the problem of syntactic and semantic analysis - one is a component of judgment or decision (whether a given string of words is a possible sentence or not), and the other is a component of representation or interpretation (deciding how the pieces of the sentence relate to each other and what they mean). In speech understanding, the former function is especially important. This judgmental function is critical in distinguishing possible word sequences that a speaker might have uttered from mere random sequences of words that happen to match the acoustic input. Without this ability to discriminate well-formed, meaningful sentences from "word salad", a speech understanding system would frequently (perhaps even usually) produce interpretations of the input that are incomprehensible.

Rules for expressing syntactic constraints on possible sentences can be expressed in several formal grammar models, such as context-free phrase structure grammars, transformational grammars (Chomsky 1965), and augmented transition network (ATN) grammars (Woods 1970). A discussion of various grammar models and parsing methods is given in Woods (1975). The ATN model is well suited both to expressing sophisticated grammars of natural language and to efficient computational use. In BBN's HWIM speech understanding system, (Woods et al. 1976), a parser was developed that can use an ATN to parse (from the middle out) an isolated fragment of an utterance and determine whether such a fragment is a possible fragment (not necessarily a well-formed constituent) of some complete sentence. Moreover, this algorithm could predict possible words and syntactic classes that could be used to extend such a fragment.

Semantic constraints on possible utterances (i.e., constraints that they be meaningful as well as grammatical) can be expressed by formal semantic interpretation rules (Woods 1978). They can also be expressed as an ATN - either in a combined syntactic/semantic/pragmatic ATN (as in HWIM) or as a separate ATN in a cascade of ATN transducers (Woods 1980). Since space here does not permit a full treatment of semantic rules and semantic interpretation, the reader is referred to the above two references for further details.

3. Theories, Monitors, Notices, and Events - A Computational Framework for Perception

The BBN speech understanding system (Woods et al. 1976; Wolf and Woods 1977) has evolved within a general framework for viewing perceptual processes. Central to this framework is an entity called a theory. A theory represents a particular hypothesis about some or all of the sensory stimuli that are present. Perception is viewed as the process of forming a believable coherent theory which can account for all the stimuli. This is arrived at by successive refinement and extension of partial theories until a best complete theory is found.

In general, a high-level perception process requires the ability to recognize any member of a potentially infinite class of perceptible objects that are constructed out of elementary constituents according to known rules. That is, the object perceived is generally a compound object, constructed from members of a finite set of elementary constituents according to some kind of well-formedness rules. These elementary constituents, as well as the relationships among them that are invoked in the well-formedness rules, must be directly perceptible. Thus, a perceptual system must incorporate some basic epistemological assumptions about the kinds of things that it can perceive and the rules governing their assembly. The well-formedness rules can be used to reject impossible interpretations of the input stimuli, and may also be useable to predict other constituents that could be present if a given partial theory is correct.

This perception framework assumes mechanisms for using subsets of the input stimuli to form initial "seed" hypotheses for certain elementary constituents (stimulus-driven hypothesization) and mechanisms for deriving hypotheses for additional compatible constituents from a partial theory (theory-driven, or predicted, hypothesization*). It also assumes mechanisms for verifying a hypothesis against the input stimuli and evaluating the well-formedness of a compound hypothesis to assign it some measure of quality and/or likelihood. A theory may therefore be thought of as a hypothesis that has been evaluated in this way and assigned a measure of confidence.*

In the case of speech understanding, a theory can range from an elementary hypothesis that a particular word is present at a particular point in the input (a word match) to a complete hypothesis of a covering sequence of words with a complete syntactic and semantic interpretation. (In general, a theory can be a set of compatible word hypotheses with gaps between them and with partial syntatic and semantic interpretations.) A partial theory may be able to generate predictions for appropriate words or classes of words either adjacent to the words already hypothesized, or possibly elsewhere in the utterance.

Predictions are dealt with in our computational framework by two kinds of devices: monitors, which are passively waiting for expected constituents, and proposals, which are elementary hypotheses that are to be evaluated against the input. Proposals result in actively seeking stimuli that would verify them, while monitors passively wait for such hypotheses to be formed. The functioning of monitors assumes that there is an organizing structure into which all derived partial hypotheses are placed as they are discovered and that the monitors can essentially set "traps" in this structure for the kinds of events that they are watching for. This is to be contrasted with continuous parallel evaluation of special processes (frequently called "demons") to watch for expected patterns in the input stream. Monitors perform no computation until and unless some other process makes an entry of the kind they are waiting for in some data structure.

The functioning of monitors is illustrated by an early speech understanding system at BBN dealing with concentrations of chemical elements in lunar rocks. There, for example, a word match for "concentration" would set monitors on the concept nodes for SAMPLE and CHEMICAL ELEMENT in a semantic network. If a word such as "Helium" was subsequently found anywhere else in the utterance, a check in the semantic network starting with Helium would lead to the superset category CHEMICAL ELEMENT where it would wake up the monitor from "concentration", thus detecting the coincidence of a detected hypothesis and a predicted hypothesis (Nash-Webber 1975).

* Our notion of stimulus-driven hypothesization is essentially the same as that of "bottom-up" processing referred to in many discussions of such processes. However, our notion of theory-driven hypothesization is slightly different from the sense usually given to "top-down" processing in that it does not necessarily imply any global ("topmost") hypotheses, but only predictability by some other hypotheses, which may itself have been derived "bottom-up". The terms "top-down" and "bottom-up" in this sense come from the literature on formal parsing algorithms.

When a monitor is triggered, an event is created calling for the evaluation of a new hypothesis and the creation of a new theory if the hypothesis is not rejected. In general, a number of events are competing for service by the processor at any moment. In human perception, there may be full parallel processing of such events, but in a serial machine, these events must be queued and given processing resources on the basis of some priority ordering. (Even in human perception, there is probably some sort of priority allocation of resources, since various kinds of interference can occur.) In our computational framework, events are maintained on a queue in order of priority, the top event being processed at each step.

The processing of an event can result in new proposals being made, new monitors being set, and existing monitors being triggered to produce new events. Since so much hinges on the event chosen for processing, a major issue is that of assigning priorities to events in order to find the most likely interpretation of the input. In the new BBN system, priority scores are assigned on the basis of Bayesian estimates of the probabilities of the competing theories, and certain control strategies and priority scoring metrics can be guaranteed to discover the most probable interpretation of the input.

4. Control Strategies

The above discussion leaves open issues such as when should initial one-word "seed" theories be formed, how many should be considered, should all seeds be worked on in parallel, etc. These issues we refer to as control issues. They have been critically important in computerized speech understanding systems. In the BBN system, for example, there are a variety of different control stategies that all fit within the above paradigm. Figure 1 illustrates one class of strategies in which seeds are formed anywhere in the utterance that sufficiently salient word matches are found. The figure shows the seed events formed as a result of an initial scan of an utterance for high likelihood word matches anywhere in the utterance. Each theory is assigned a score expressing its likelihood of being correct (actually a logarithm of the ratio of the likelihood of the acoustic evidence given the theory over the a priori likelihood of that evidence occurring independently). The region of the utterance covered by the theory is indicated by specifying its left and right boundary positions in a list of potential boundary positions (the left end of the utterance is numbered 0 and in this case the right end is numbered 18). The exclamation marks indicate the theories that are actually part of the correct interpretation.

For this general class of control strategies, referred to as "middle-out" or "island-driven" strategies, theories are grown by starting with a seed word, asking a higher-level linguistic component to predict categories of words that can occur on either side of it, asking a lexical retrieval component to find the best matching words in those categories on the appropriate sides, and generating events for each such word found to extend the theory by adding that word. Thus, events will be placed on the event queue to add words both on the left and on the right ends of given theories. These "new word" events will compete with each other and with the remaining seed events on the basis of score to determine which event will be processed next, causing the processor sometimes to continue adding words to a given theory and at other times to shift its processing to a different competing theory.

Figure 2 shows the sequence of theories that are formed as a result of this process, starting with the event queue of Figure 1. (Brackets in the figure indicate theories that include the hypothesis that the left or right ends of the utterance have been reached. A number in parentheses after a theory number is the number of a preceding theory from which the indicated theory was formed by the addition of a new word.) Notice that the final theory is developed in this case by working independently on two different portions of the utterance starting from the seeds "shown" and "trips". The final theory in Figure 2 is in fact derived from a kind of event called a collision event which combines the theories "show me" and "trips" when they both notice the word "her" filling the gap between them. This event is formed during the processing of theory 13, although its score is such that it does not reach the top of the queue until theory 23.

Figure 3 shows the seed theories for a hybrid strategy in which seeds are started within a bounded distance from the left end of the utterance, and are grown right-to-left until they reach the left end, after which the remainder of the processing is left-to-right. Figure 4 shows the sequence of theories developed in the course of understanding this utterance using the hybrid strategy. The basically left-to-right nature of the hybrid strategy, except for a bounded initial delay in getting started, seems to be a reasonable possibility for a model of human speech understanding, since it is clear that human processing of speech does not involve the buffering of a complete sentence before understanding begins.

5. Priority Scoring

The scoring assigned to a theory by the summation of individual word scores (essentially the log probability of its words being correct) we refer to as the quality score of the theory. We distinguish from this a possibly separate score called the priority score, which is used to rank order events on the event queue to determine the order in which they are to be processed. In early versions of HWIM, we used the quality score itself as the priority score. However, we have developed several algorithms with interesting

#	SCORE	REGION	THEORY
1	0	11-14	ADD
2	0	4-7	NEED
3	-.455	0-3	SHOW !
4	-.605	12-17	TRIP
5	-.727	1-5	ROME
6	-.769	8-11	THERE
7	-1.25	12-18	TRIP-S !
8	-1.47	0-5	SHELLY
9	-1.65	15-17	END
10	-1.72	11-14	AND
11	-1.73	1-5	ANN
12	-1.74	0-5	CHEYENNE
13	-2.19	8-14	BERT
14	-2.26	2-6	ANY
15	-2.82	0-5	SOME

+15 ADDITIONAL EVENTS

Fig. 1. Seed events for middle-out strategy

```
THEORY 1                                 ADD
THEORY 2                    NEED
THEORY 3                 SHOW !
THEORY 4(3)              [SHOW !
THEORY 5(4)              [SHOW ALL
THEORY 6(3)              SHOW ALL
THEORY 7                                 TRIP
THEORY 8                  ROME
THEORY 9(4)              [SHOW ME !
THEORY 10(3)             SHOW ME !
THEORY 11(7)                     HER TRIP
THEORY 12                           TRIP-S !
THEORY 13(12)                       TRIP-S ] !
THEORY 14(13)                    HER TRIP-S ] !
THEORY 15(12)                    HER TRIP-S !
THEORY 16                SHELLY
THEORY 17(16)            [SHELLY
THEORY 18                [SHOW ME HER TRIP
THEORY 19                SHOW ME HER TRIP
THEORY 20(11)                    HER TRIP IS
THEORY 21(20)                    HER TRIP IS ]
THEORY 22(20)                   OF HER TRIP IS
THEORY 23(9,13)          [SHOW ME HER TRIP-S ] !
```

Fig. 2 Theories formed for middle-out strategy

#	SCORE	REGION	THEORY
1	3.53	1-5	WHO
2	1.92	3-6	WE !
3	0.0	0-1	-PAUSE- '
4	-2.43	2-3	A
5	-3.24	5-10	ELEVEN
6	-4.32	5-9	IRAQ
7	-5.36	1-3	HER
8	-6.00	1-4	WHOLE
9	-6.18	1-5	DO !
10	-6.21	3-6	WERE
11	-6.53	3-7	WORK
12	-6.85	1-4	HIS
13	-7.00	1-5	HOW
14	-7.12	1-6	HAWAII
15	-7.21	3-6	WHERE

+39 ADDITIONAL EVENTS

Fig. 3. Seed events for hybrid strategy

```
THEORY 1             WHO
THEORY 2(1)          ⌈WHO
THEORY 3               WE !
THEORY 4(3)          DO WE !
THEORY 5(4)          ⌈DO WE !
THEORY 6(3)            ⌈WE !
THEORY 7(5)          ⌈DO WE HAVE !
THEORY 8(7)          ⌈DO WE HAVE A !
THEORY 9             -PAUSE- !
THEORY 10(9)         ⌈-PAUSE- !
THEORY 11(3)           ARE WE
THEORY 12(8)         ⌈DO WE HAVE A SURPLUS !
THEORY 13(12)        ⌈DO WE HAVE A SURPLUS⌉ !
```

Figure 4. Theories formed for hybrid strategy

theoretical properties using priority scores that are derived from, but not identical with, the quality score. The first measures the difference between the particular quality score for a theory and an upper bound on possible quality scores for any theory covering the same portion of the utterance. We call this the shortfall score, and it can be shown that using the shortfall score as a priority score under appropriate conditions guarantees finding the best scoring interpretation of the unput utterance (Woods et al. 1976, Woods 1977). Using the quality score itself as a priority score does not guarantee this. Other priority scores are obtained by dividing either the quality score or the shortfall score by the time duration of the island to give quality density and shortfall density scoring, respectively. Since a fairly complete derivation of the shortfall density scoring strategies together with proofs of their theoretical properties is given in Woods (1977), we will present here only a brief recapitulation of the strategies, and discuss differences we have observed between them.

5.1 **Shortfall Scoring**

The shortfall score measures the amount by which the quality score of a theory falls an upper bound on the possible score that could be achieved on the same region. When shortfall scoring is being used, a MAXSEG profile is constructed having the property that the score of a word between boundaries i and j will be less than or equal to the area under the MAXSEG profile from i to j (call this latter the MAXSCORE for the region from i to j). The shortfall score for a theory is then computed as the sum over all the word matches in the theory of the difference between the score of the word match and the MAXSCORE for the same region. The preferred theory is the one with the smallest magnitude of shortfall.

The MAXSEG profile can be constructed incrementally by adding to the profile whenever a word match is found whose score is not bounded by it. Whenever the score of a word match exceeds the MAXSCORE for its region, the excess score is distributed over the region to raise its MAXSCORE to that of the word match. In HWIM, an initial profile is constructed during the initial scan for seed words and this profile is substantially correct. Occasionally, a word match is found later which raises the profile, and in this case all events overlapping the changed region are rescored.

In order to satisfy the theoretical claims of the algorithms, the way in which the excess score of a word match is distributed to raise the MAXSEG profile does not matter. However, it is desirable to do it in such a way as to minimize the amount by which the shortfall of other words that overlap the region is raised. Our current algorithm is to distribute the excess score over the segments covered by the word match that are not already bounded by the profile and to divide it in proportion to the durations of the segments. Other distribution algorithms are possible, some of which have been tried.

This one is better than some, but there are probably better strategies to be found. Keeping the MAXSEG profile as low as possible while still satisfying the upper bound condition is important since excessively conservative upper bounds translate directly into an unnecessary increase in the breadth-first nature of the search, requiring more events to be processed before finding the chosen interpretation.

The theoretical characteristics of the shortfall scoring algorithm are that if the words are returned by the Lexical Retrieval component in decreasing order of quality and events are processed in order of increasing magnitude of shortfall (plus a few other assumptions, documented in Woods (1977)), then the first complete spanning interpretation found will be the best scoring interpretation that can be found by any strategy. We refer to this condition as "completeness" (a more traditional term is "admissibility"). For speech understanding applications, completeness is a desirable property, but not necessarily essential if the cost of its attainment is too great. Shortfall scoring has the property of being complete without searching the entire space. Its completeness proof depends only on the fact that when the first complete spanning theory is found, all other events on the queue will already have fallen below the ideal maximum score by a greater amount. Thus, the result does not depend on the scores being likelihood ratios, nor does it make any assumption about the nature of the grammar (e.g., that it be a finite-state Markov process) provided a parser exists that can make the necessary judgements. The completeness also does not depend on the order of scanning the utterance - it is satisfied both for middle-out and for left-to-right strategies.

5.2 Density Scoring

Another type of priority scoring is density scoring. Here the score used to order the event queue is some basic score divided by the duration of the event. Conceptually, we can think of this priority scoring metric as predicting the potential score for the region not covered by a theory to be an extrapolation of the same score density already achieved. (In these terms, the shortfall strategy can be thought of as predicting that the upper bound for the uncovered region will be achieved.) Unlike the shortfall scores, density scores can get bad and then get better again as new words are added to a theory. Hence, the density score is certainly not guaranteed to be an upper bound of the expected eventual score. However, it has another interesting property : in exactly those cases where it does not bound the eventual score, there is a word to be added somewhere else that has a better score density and whose score density does bound the eventual score. This arises from the property of densities that the density of two regions combined will lie between the densities that they each have. It turns out that this alone is not sufficient to guarantee completeness for a density scoring strategy, since it is still possible for the density score starting from the best correct seed to fall below that of some other less-than-optimal spanning theory before it can be extended to a complete theory itself. However, with the addition of a facility for combining islands that start from separate seeds when they collide with each other, the density scoring strategy working middle-out from multiple seeds can be shown to be complete. Again, density scoring does not depend on any assumptions about the basic scores to which it is being applied other than that they be additive (and capable off division). Hence, the density method can be applied to either the original quality score or to a shortfall score. The combination of the two methods in a shortfall density strategy seems to be more effective than either shortfall or density scoring alone.

6. Comparison of Speech Understanding Systems

6.1 BBN HWIM

The admissible strategies discussed above are only some of the control strategy options implemented in the BBN HWIM speech understanding system. In addition there are a large number of strategy variations that result in approximate strategies, including strictly left-to-right strategies and "hybrid" strategies that start near the left end of an utterance and work left and then right. For reasons of time and resource limitations, our final test run of the HWIM system was made using one of the approximate strategies. Subsequently, a much smaller experiment was run to compare various control strategies on a set of ten utterances at random from the larger set. Although this sample is much too small to be relied on, the results are nevertheless suggestive. For two comparable experiments using our best left-to-right method (left-hybrid shortfall density) and our best island-driven method (shortfall density with ghosts, island collisions, and direction preference), both with a resource limitation of 100 theories and without using a facility for analysis-by-synthesis word verification, the results were as follows:

	LHSDNV	SD+GCD
Correct interpretation	6	5
Incorrect interpretation	2	0
No interpretation	2	5
Average number of theories evaluated	50.7	75.5

That is, the left-to-right strategy found the best (and in these cases the correct) interpretation within the resource limitation in 6 of the 10 cases, while the island-driven strategy found only 5 (not necessarily a significant difference for this size sample). On the other hand, the left-to-right method misinterpreted two additional utterances with no indication to distinguish them from the other 6. If this strategy were used in an actual application with comparable degrees of acoustic degradation (e.g. due to a noisy environment), the system would claim to understand 80% of its utterances, but would actually misunderstand 25% of those. The island-driven strategy, on the other hand, would only claim to understand 50% of the utterances, but would misunderstand a negligible fraction.

The island-driven strategy in the above experiments expanded only 50% more theories (and incidentally used only 30% more cpu time) than did the left-to-right strategy. Although, as we said before, this test set is much too small to draw firm conclusions, the success rate of the two methods are not much different, excpet that the island-driven shortfall density method is clearly less likely to make an incorrect interpretation. Moreover, the numbers of theories considered and the computation times are not vastly different. If one considers proposals to improve the performance of left-to-right strategies by having them continue to search for additional interpretations after the first one is found (and thus take the best of several), then the time difference shown above could easily be reversed and there would still be no guarantee that the best interpretation found would be the best possible.

6.2 **DRAGON**

The DRAGON system (Baker 1975) is the only other speech understanding system in the ARPA project that provides a guaranteed optimum solution. It does this by using a dynamic programming algorithm that depends on the grammar being a Markov process (i.e. a finite-state grammar). It operates by incrementally constructing, for each position in the input and each state in the grammar, the best path from the beginning of the utterance ending in that state at that position. The computation of the best paths at position i+1 from those atposition i is a relatively straightforward local computation, although the number of operations for each such step, for a grammar with n states, is n times the branching ratio (i.e. the average number of transitions with non-zero probability leaving a state). DRAGON performs such a step for each 10 millisecond portion of the utterance using a state transition that "consumes" an individual allophonic segment of a phoneme.

The optimality of the solution found by this algorithm depends on the property of finite state grammars that one sequence of words (or phonemic segments) leading to a given state * is equivalent to any other such sequence as far as compatibility with future

* I am using the term "state" a little casually here in roughly the sense that it is used in an ATN grammar (Woods 1970). If one takes the condition of having equivalent future predictions as the definition of a "state" of a grammar, then what the finite-state grammar does is guarantee that there are only a finite number of such states, which can therefore be enumerated and named ahead of time. For a more general grammar, the number of such states is open-ended.

predictions is concerned (regardless of the particular words used). It is this property that permits the algorithm to ignore all but the best path leading to each state (even if competing paths score quite well), and therefore permits it to find the best solution by progressively extending a bounded number of paths across the utterance from left to right. (This is a very attractive property, although in this case it requires one such path for each state in the grammar.) For more general grammars, where there may be context-sensitive checking between two different parts of the utterance (e.g. number agreement between a subject and a verb), the best path leading to a given state at a given position may not be compatible with the best path following it. In this case, second best (and worse) paths must be considered in order to guarantee finding any complete paths at all (much less an optimum).

Although only applicable to finite-state languages, DRAGON's dynamic programming method has the advantage of taking a relatively constant amount of time from utterance to utterance, being simple to compute, and guaranteeing to obtain the optimal solution. The only difficulty is that for a large number of states in the grammar (e.g. thousands for a reasonable size grammar) the amount of computation required is expensive. Except to the extent that the finite-state grammar permits one to eliminate from consideration any path that is not the best one leading to its state, the algorithm exhaustively enumerates all other possibilities.

Although DRAGON's scores are estimates of probabilities of interpretations, its guarantee of optimality does not depend on that, but only on the fact that its grammar is finite-state and that therefore it suffices to carry a record of the best path leading to each state. The same dynamic programming algorithm can be applied at the level of phonemes or words, and can be generalized to apply to an input lattice such as the BBN segment lattice (Woods et al. 1976)

6.3 HARPY

The CMU HARPY system (Lowerre 1976) is a development of the DRAGON theme which gives up an absolute guarantee of optimality in exchange for computation speed. Like DRAGON, it takes advantage of the unique characteristic of finite-state grammars cited above, so that only the best path leading to a given state need be considered. However, it uses an adaptation of the dynamic programming algorithm in which not all of the paths ending at a given position are constructed. Specifically, at each step of the computation, those paths scoring less than a variable threshold are pruned from further consideration. (In the Itakura-metric version of the system, setting this threshold at 1/100000 of the score of the best path at that point was reported to give the best performance.) This gives an algorithm that carries a number of paths in parallel (the number varying and depending on the number of competitors above the threshold at any given point) but is not exhaustive. If the threshold is chosen appropriately, the performance closely approximates that of the optimal algorithm, although there is a tradeoff between the speed efficiency gained and the chances of finding a less than optimal path.

In 1976, the HARPY system had the best demonstrated performance statistics of any continuous speech understanding system. However, it derived this performance in large part from the use of a highly constraining (and advantageously structured) finite-state grammar. This grammar has an average branching ratio of approximately 10, and characterizes a non-habitable, finite set of sentences, with virtually no "near miss" sentence pairs included. For example, "What are their affiliations" is in the grammar, but no other sentences starting with "What are their" are possible. The only two sentences starting with "What are the" are "What are the titles of the recent ARPA surnotes," and "What are the key phrases." These three sentences will almost certainly find some robust difference beyond the initial three words that will reliably tell them apart. Similarly, the grammar permits sentences of the form "We wish to get the latest forty articles on <topic>," but one cannot say a similar sentence with "I" for "we", "want" for "wish", "see" for "get", "a" for "the", "ten" for "forty", or any similar deviation from exactly the word sequence given above.) Most of HARPY's grammar patterns (such as the last one) consist of a particular sentence with one single open category for either an author's name or a topic. A large number of them are particular sentences with no open categories (like the first three above).

The HARPY algorithm makes no guarantee that the correct path will not be pruned from consideration if it starts out poorly, but at least for the structure of HARPY's current grammar (most of whose sentences start with stressed imperative verbs or interrogative pronouns), the correct interpretation is usually found.

The HARPY technique (or variations of it) seems to be the algorithm of preference at present for applications involving carefully structured artificial languages with finite-state grammars and small branching ratios (on the order of 10 possible word choices at each position in an utterance). However, it does not conveniently extend to larger and more habitable grammars. This is due to a number of factors, including: the combinatorics of expanding them into a finite-state network (the branching ratio 10 grammar on which its best performance is reported is about the largest HARPY could hold in its memory), the approximations necessary to represent such a grammar as a finite-state network (most such grammars are at least context-free and usually context-sensitive - so that finite-state approximations necessarily accept sentences that the original grammar doesn't or fail to accept some that it does), and the difficulty of dealing with dynamically changing situations such as constraints on utterances that depend on previous sentences.

Neither the DRAGON nor the HARPY system use density normalization or any method to attempt to estimate the potential score that is achievable on the as yet unanalyzed portion of the utterance. Such normalization is not necessary, since they both follow paths in parallel, all of which start and end at the same point in the utterance, and therefore never have to compare paths of different lengths or in different parts of the utterance. Again, it is worth emphasizing that the ability of these algorithms to keep the number of paths that need to be considered manageable depends on the unique characteristic of finite-state languages that requires only the best path to each state to be considered.

6.4 IBM

A group at IBM (Bahl et al. 1976) has a speech understanding system based on Markov models of language, which has implemented two control strategies: a Viterbi algorithm (essentially the same dynamic programming algorithm used by DRAGON) and a "stack decoder", a left-to-right algorithm with a priority scoring function that attempts to estimate the probability that a given partial hypothesis will lead to the correct overall hypothesis. The latter apparently does not guarantee the optimal interpretation, but somehow is reported as getting more sentences correct than the other (a circumstance I don't fully understand, but which can happen if there are acoustic-phonetic scoring errors such that the best scoring interpretation is not correct).

Recent experiments with an improved version of one of the IBM systems (incorporating the CMU technique of bypassing a phonetic segmentation to do recognition on fixed length acoustic segments (Bahl et al. 1978) reported performance on the same grammar used in the HARPY system (the "CMU-AIX05 Language") of 99% correct sentence understanding. (This performance is based on recordings in a noise-free environment, however, compared to a rather casual environment for the CMU results). They also report performance of 81% correct sentence understanding on a more difficult, but still small branching ratio, finite-state grammar (their "New Raleigh Language"). Both of these results were obtained in experiments with the system trained for a single speaker and tested on that same speaker. Performance of the system when tested with a difficult speaker is significantly less.

6.5 Hearsay II

The Hearsay II System (Lesser et al. 1975) permits the kind of generalized middle-out parsing described in this paper, and does so for context-free grammars (although apparently not for context-sensitive or more powerful grammars). Moreover, it has a capability for a kind of island collisions (Erman--personal communication). However, its design philosophy specifically rejects the use of an "explicit control strategy" as "inappropriate" (because it "destroys the data-directed nature and modularity of knowledge source activity" (Hayes-Roth & Lesser 1976)). Its scoring function for hypotheses, which its authors refer to as the "desirability" of a KS (knowledge source), is an ad hoc combination of functions reflecting intuitive notions of "value", "reliability", "validity", "credibility", "significance", "utility", etc. Specifically, they state: "the desirability of a KS invocation is defined to be an increasing function of the following variables: the estimated value of its RF (an increasing function of the reliability of the KS and the estimated level, duration and validity credibility of the hypothesis to be created or supported); the ratio of the estimated RF value to the minimum current state in the time region of the RF; and the probability that the KS invocation will directly satisfy or indirectly contribute to the satisfaction of a goal as well as the utility of the potentially satisfied goal" (Hayes-Roth & Lesser 1976).

They go on to say that the above is not "complex enough" to "provide precise control in all of the situations that arise", and proceed to describe various further elaborations, all of which are vague as to exactly what the system does.

Although it is extremely difficult to tell from the available published descriptions exactly what Hearsay II does, the fact that the "desirability" of a KS invocation is an increasing function of its duration definitely rules out any interpretation of it as implementing the density method. The above allusion to the "current state in the time region of the RF" refers to a parameter that for each point t in the utterance specifies the maximum of the "values" of all hypotheses "which represent interpretations containing the point t". This "state" function at first glance seems similar to the maxseg profile used in the shortfall algorithms (and indeed was what caused me to start thinking along those lines), but in actuality it is quite different. Instead of being an estimate of the maximum possible portion of a score that can be attributed to a segment, Hearsay's state is the maximum total score of any hypothesis found so far that covers it (recall that such scores increase with length of the theory). Its contribution to the desirability of a hypothesis is the ratio of the "value" of that hypothesis to the smallest value of the state parameter in its region.

Since the smallest state value in the region of a hypothesis will always be at least as great as that of the hypothesis being valued (each state is the max value of all covering hypotheses), this ratio is always less than or equal to one, and is strictly less only when every portion of the region covered by the hypothesis has some better covering hypothesis (although not necessarily a single hypothesis that covers the whole region). Consequently, the "state" component of the score has the effect of inhibiting a hypothesis that at every point has a better competitor. Since the values of hypotheses grow with the length of the region covered, the effect will be that hypotheses that get big early will inhibit alternative hypotheses on the regions they cover. With shortfall scoring, on the other hand, the tendency is for big hypotheses to pick up additional shortfall and increase the likelihood of a shift to a competing hypothesis that might ultimately get a better score (this is what makes it an admissible algorithm). Hearsay II's use of the "state" parameter, is more reminiscent of SRI's "focus by inhibition" technique discussed below, which was found to have generally undesirable effects, although it did offset some of the costs of their island driving strategy (Paxton 1977).

In summary, it is difficult to say exactly what the Hearsay focus of attention strategy does, or how it relates to the methods presented here, except to say that it is certainly not the same as any of these methods.

A superficial comparison of the Hearsay II system performance with that of the BBN HWIM system might lead one to believe that the Hearsay II control strategy is somehow more effective. However, it is more likely that the difference in performance is due to the differences in difficulty of the two grammars or to differences in their acoustic "front end". The reported performance results of the Hearsay II system are based on the same highly constrained, branching ratio 10 grammar used by HARPY. The BBN grammar, on the other hand, is a general ATN grammar with average branching ratio (measured from hypothesis predictions in a running system) of 196, permitting a relatively habitable subset of English which includes such minimal pairs as "What is the registration fee" and "What is their registration fee". Informal conversation with members of the Hearsay II project convinces me that Hearsay II can in principle explore all the alternaties that the SD+C strategy would and would in fact explore at least these if functioning according to its design philosophy of finding a first interpretation and then exploring further any hypotheses that could produce something better.

6.6 The SRI Experiments

At SRI, Paxton (1977) has performed a number of experiments on control strategy options, using a simulated word matching component based on performance statistics of the SDC word matching component to which a speech understanding system at SRI was originally intended to be coupled. Paxton's system is well-documented, and contains a number of interesting and well-done capabilities. He has worked out a very clean representation of the SRI grammar as a collection of small ATN networks (although he doesn't call them that) which do not have the directional left-to-right orientation that conventional ATN's do and in which the association of augments with transitions is more systematized and less procedural. The capabilities of this system for syntactic/semantic/pragmatic constraint are comparable in power to that of HWIM's general ATN grammar, and in several respects the notations used are cleaner and more perspicuous than one usually finds in a conventional ATN. Moreover, the implementation of these grammars contains some very elegant efficiency techniques. The system has a capability for middle-out parsing making use of the semantic/pragmatic augments in the grammar, although it doesn't seem to have a capability for island collisions and doesn't construct islands for arbitrary sentence fragments.

In terms of the control strategy framework set up in this paper (as opposed to the terms that he himself uses), Paxton's system makes a distinction between a quality score for a hypothesis and a priority score for an event, although the kinds of hypotheses and events that his system creates are somewhat different than those above. One way of viewing his system in the terms presented here is that his hypotheses are always partially completed constituents (what he calls "phrases"), which can make predictions for the kinds of words or constituent phrases that they can use. These phrases are incorporated into a structure called a "parse net" in which explicit "producer" and "consumer" links associate such hypotheses to each other, but partially completed phrasess are not combined into larger sentence fragments corresponding to our notion of islands (which can be partial at several levels of phrase structure). His events are of two types: operations to look for a word or words at a point (what he calls a "word task", comparable to our proposals to the lexical retrieval component), and events to create such predictions from a phrase (what he calls a "predict task"). Every phrase is implicitly an event for a predict task, and he has a special data type called a "prediction" to represent events for word tasks.

Whereas HWIM, when it processes a hypothesis, will always make all predictions, call the lexical retrieval component to find all matching words, and create word events for each such found word, Paxton's system breaks this cycle up differently. His system schedules separate events for each of the individual word predictions generated by a hypothesis, and whenever a word or completed phrase is found he distributes it immediately to all its "consumers" without waiting. (This difference is probably motivated by his lack of a word matcher that could efficiently find the best matching words at a given position without exhaustively considering each word in the dictionary.)

Paxton's system makes no attempt to guarantee the best interpretation, nor does it stop with the first complete interpretation it finds. Rather it runs until one of several stopping conditions is satisfied (such as running out of storage), after which it takes the best interpretation that it has found so far.

Paxton performed a systematic set of experiments varying four control strategy choices, which he called "focus by inhibition", "map all at once", "context checking", and "island driving". The first was a strategy for focusing on a set of words that occur in high scoring hypotheses and decreasing the scores of all tasks for hypotheses incompatible with those words.

The "map all at once" strategy referred to a "bottom up" lexical retrieval strategy that found all possible words at a given point and ranked them taking their word mapper scores into account, rather than proposing such words one at a time in the order in which their proposing hypothesis ranked them (essentially ranking such words according to a priori preferences assigned by the grammar).

"Context checking" referred to a technique of assigning a priority score to predictions of a partial phrase on the basis of a heuristic search for the best possible combinations of higher level constituents that can use that phrase, rather than by basing such priority scores solely on the local quality of the partial phrase alone. (This mechanism gives part of the effect of our use of theories that include arbitrary fragments of a sentence that may cross several levels of phrase boundary, but does not apparently permit a fragment that has incomplete phrases at both ends to be prioritized as a whole. It assigns the resulting priority score just to the phrase doing the prediction without apparently remembering the context that justified this score).

"Island driving", in Paxton's system, referred to the use of a middle-out strategy that looked for a best word somewhere in the utterance to start a seed, and if all hypotheses from that seed scored badly enough would look for another such seed, and so on. However, his system contained none of the features such as island collisions, ghosts, preferred directions, shortfall, or density scoring techniques discussed in this paper, although it may have had something amounting to an absolute direction preference (the documentation is not totally clear on whether both ends of an island can be worked on independently). Hence its version of island driving seems to have all of the disadvantages of a middle-out strategy with almost none of the compensating advantages.

The experiments indicated that the "main effects" of focus by inhibition (i.e., the net effects averaged over all combinations of other strategy options) wwere negative both in accuracy of the recognition and in number of events processed, and that the main effects of mapping all at once and context checking were positive (the former was more expensive in run time in their system, but might not have been with a suitable lexical retrieval component such as that of HWIM). All three of these experiments showed a statistically significant effect. In addition, the main effect of their island driving feature was found to be negative in time and accuracy, although the result was not statistically significant "because of a large interaction with sentence length". Specifically, Paxton found that island driving improved performance for short utterances, but decreased performance for longer ones, largely due to exceeding the storage limitations before finding the best interpretation. Consequently, it is possible that the implementation of some of the features described in this paper might have improved the performance of the island driving strategy sufficiently to gain a net improvement.

Paxton's results with the focus of inhibition strategy reflect what seems to have been a common experience of the various speech understanding groups in the ARPA project. Although it seems natural to expect that some word match scores should be good enough that they could be considered correct, thereby eliminating attempts to find alternatives to them, in fact all attempts to implement such an intuition seem to have lead to at best indifferent results and usually to positive degradation. In retrospect, the fact that perfect matches of other words or short word sequences can occur by accident in completely accurate transcriptions of sentences should suggest that there is no magic threshold above which one can consider a given hypothesis correct without verifying its consistent extension to a complete spanning theory. It seems, therefore, that the absolute value of the local quality score is not what matters in deciding the most likely interpretation. The relative scores of competing hypotheses are more relevant, but what really counts is the eventual quality of the complete spanning theory.

Acknowledgment

Much of the work described here was supported by the Advance Research Projects Agency of the Department of Defense and was monitored by ONR under Contract No. N00014-75-C-0533.

References

Bahl, L.R., Baker, J.K., Cohen, P.S., Dixon, N.R., Jelinek, F., Mercer, R.L. and Wilverman, H.F. (1976). "Preliminary Results on the Performance of a System for the Automatic Recognition of Continuous Speech", Conference Record, IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP-76, Philadelphia, Pa., April, 1976.

Bahl, L.R., Baker, J.K., Cohen, P.S., Cole, A.G., Jelinek, F., Lewis, B.L. and Mercer, R.L. (1978). "Automatic Recognition of Continuously Spoken Sentences from a Finite State Grammar". Conference Record, 1978 IEEE Int'l Conf. on Acoustics, Speech and Signal Processing, Tulsa, Okla., April, 1978, IEEE 78CH1285-6 ASSP.

Baker, J.K. (1975). "The DRAGON System -- An Overview", IEEE Trans. Acoustics, Speech and Signal Processing, February, Vol. ASSP-23, No. 1, pp. 24-29.

Chomsky, N. (1965). "Aspects of the Theory of Syntax", Cambridge, Mass.: MIT Press.

Hayes-Roth, F. and Lesser, V.R. (1976). "Focus of Attention in a Distributed-Logic Speech Understanding System", Conference Record, IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP-76, Philadelphia, Pa., April, 1976.

Klatt, D.H. and Stevens, K.N. (1971). "Strategies for Recognition of Spoken Sentences from Visual Examination of Spectrograms", BBN Report No. 2154, Bolt Beranek and Newman Inc., Cambridge, Mass.

Lea, Wayne (1980). "Trends in Speech Recognition", Prentice-Hall, Englewood Cliffs, N.J.

Lesser, V.R., Fennell, R.D., Erman, L.D. and Reddy, D.R. (1975). "Organization of the Hearsay II Speech Understanding System", IEEE Trans. Acoustics, Speech and Signal Processing, February, Vol. ASSP-23, No. 1, pp. 11-24.

Lowerre, Bruce T. (1976). "The HARPY Speech Recognition System", Technical Report, Department of Computer Science, Carnegie-Mellon Univ., April, 1976.

Nash-Webber, B.L. (1975). "The Role of Semantics in Automatic Speech Understanding", in Representation and Understanding: Studies in Cognitive Science, D. Bobrow and A. Collins (eds.), New York: Academic Press.

Newell, A. et al. (1973). "Speech Understanding Systems: Final Report of a Study Group", Amsterdam : North Holland/American Elsevier.

Paxton, W.H. (1977). "A Framework for Speech Understanding", Stanford Research Institute Artificial Intelligence Center, Technical Note 142, June 1977.

Wanner, E. (1973). "Do We Understand Sentences from the Outside-in or from the Inside-out?", Daedalus, pp. 163-183, Summer.

Wolf, J.J. and Woods, W.A. (1977). "The HWIM Speech Understanding System", Conference Record, IEEE International Conference on Acoustics, Speech and Signal Processing, Hartford, Conn., May, 1977.

Woods, W.A. (1970). "Transition Network Grammars for Natural Language Analysis", Communications of the ACM, Vol. 13, No. 10, October, 1970, pp. 591-606.

Woods, W.A. (1975). "Syntax, Semantics and Speech", in D.R. Reddy (ed.), Speech Recognition : Invited Papers at the IEEE Symposium, New York: Academic Press. (Also as BBN Report No. 3067).

Woods, W.A. (1977). "Theory Formation and Control in a Speech Understanding System with Extrapolations towards Vision", Proc. of Workshop on Computer Vision Systems, University of Massachusetts, Amherst, June.

Woods, W.A. (1978). "Semantics and Quantification in Natural Language Question Answering", in Advances in Computers, Vol. 17, Academic Press, New York.

Woods, W.A. (1980). "Cascaded ATN Grammars", in American Journal of Computational Linguistics, Vol. 6, No. 1, January-March 1980.

Woods, W.A. and Makhoul, J.I. (1974). "Mechanical Inference Problems in Continuous Speech Understanding", Artificial Intelligence, Vol. 5, No. 1, pp. 73-91.

Woods, W., Bates, M., Brown, G., Bruce, B., Cook, C., Klovstad, J., Makhoul, J., Nash-Webber, B., Schwartz, R., Woolf, J. and Zue, V. (1976). "Speech Understanding Systems - Final Technical Progress Report", BBN Report No. 3438, Vols. I-V, Bolt Beranek and Newman Inc., Cambridge, Mass.