Optimal Search Strategies for Speech Understanding Control

W. A. Woods
Bolt Beranek and Newman Inc.
Cambridge, MA 02238

Abstract

This paper describes two algorithms for finding the optimal interpretation of an unknown utterance in a continuous speech understanding system. These methods guarantee that the first complete interpretation found will be the best scoring interpretation possible. Moreover, unlike other optimal strategies, they do not make finite-state assumptions about the nature of the grammar for the language being recognized. One of the methods, the density method, is especially interesting because it is not an instance of the "optimal" A* algorithm of Hart, Nilsson, and Raphael, and appears to be superior to it in the domains in which it is applicable. The other method, the shortfall method, is an instance of the A* algorithm using a particular heuristic function. Proofs of the guaranteed discovery of the best interpretation and some empirical comparisons of the methods are given. The relationship of these methods to strategies used in existing speech understanding systems is also discussed. Although presented in the speech context, the algorithms are applicable to a general class of optimization and heuristic search problems.

1. INTRODUCTION

This paper is concerned with optimal decoding strategies for continuous speech understanding. Specifically, it is concerned with control strategies governing the formation and refinement of partial hypotheses about the identity of an utterance that can guarantee the discovery of the best possible interpretation.

We assume a system that contains the following components:

a) A Lexical Retrieval component that can find the k best matching words in any region of an utterance subject to certain constraints and can be recalled to continue enumerating word matches in decreasing order of goodness (where possible constraints include anchoring the left or right end of the word to particular points in the utterance or to particular adjacent word matches). We

assume that this component is interfaced to appropriate signal processing, acoustic-phonetic and phonological analysis components as in (Woods et al., 1976), and that it assigns a "quality" score to each word match reflecting the goodness of the match.

b) A Linguistic component that, given any sequence of words, can determine whether that sequence can be parsed as a possible initial, final, or internal subsequence of a syntactically correct and semantically and pragmatically appropriate utterance, and can propose compatible classes of words at each end of such a sequence.

The HWIM speech understanding system developed at BBN (Woods et al., 1976; Wolf and Woods, 1977) has such capabilities. A control strategy for such a system must answer questions such as:

a) At which points in the utterance to call the Lexical Retrieval component, and when,

b) What number of words to ask for,

c) When to give subsequences of the results to the Linguistic component, and

d) When to recall the Lexical Retrieval component to continue enumerating words at a given point.

The goal of the control strategy is to discover the best scoring sequence of words that covers the entire utterance and is acceptable to the Linguistic component. We will consider here a particular class of control strategies which we refer to as "island-driven".

2. ISLAND-DRIVEN STRATEGIES

In an island-driven control strategy, partial hypotheses about the possible identity of the utterance are formed around initial "seed" words somewhere in the utterance and are grown into larger and larger "island" hypotheses by the addition of words to one or the other end of the island. Occasionally, two islands may "collide" by proposing and discovering the same word in the gap between them and may then be combined into a single larger island.

Each island hypothesis is evaluated by the Lexical Retrieval component to determine its degree of match with the acoustic evidence and is checked for syntactic, semantic, and pragmatic

consistency by the Linguistic component. We will refer to a partial hypothesis that has been so evaluated and checked for consistency as a "theory". The strategies that we will consider operate by successively processing "events" on an event queue, where events correspond to suspended or dormant processes that may result in the creation of theories.

The general algorithm operates as follows:

(1) An initial scan of the utterance is performed by the Lexical Retrieval component to discover the n best matching words anywhere in the utterance according to some criterion of "best" and for some value n.* An initial seed event is created for each such word and placed on the event queue. In addition, one or more continuation events, which can be processed to continue the enumeration of successively lower scoring words (regardless of position in the utterance), is created and placed on the queue. Each seed event is assigned a priority score (derived, in one of several ways to be described shortly, from the quality score that the Lexical Retrieval component gave it). Each continuation event is assigned a priority score that can be guaranteed to bound the priority score of any word that can be generated by that event (e.g., derived from the score of the last word enumerated prior to the continuation). The events are ordered on the event queue by their priority scores and are processed in order of priority.

(2) The highest priority event is selected for processing. This consists of (i) creating the corresponding theory (a one-word theory in the case of a seed event), (ii) calling the Linguistic component to check the consistency of the theory and to make predictions for words and/or word classes that can occur adjacent to it, at each end of the theory, (iii) calling the Lexical Retrieval component to enumerate the k best matching words satisfying these predictions at each end of the theory, and (iv) generating a "word" event for each such word found. A word event is an event that will add one word to a theory to create a larger theory. Continuation events are also created that will continue the enumeration of successively lower scoring words adjacent to the theory. If island-collision is permitted as an operation (island collision is a feature than can be enabled or disabled by a flag), then each word event generated is checked against an island table

* The HWIM system also has the ability to execute left-fo-right, right-to-left, and various hybrid strategies by appropriately constraining this initial search (e.g., confining it to the left end).

to see if the same word (at the same position in the input) has been proposed and found in the other direction by some theory. If so, an "island-collision" event is created that will combine the new word and the two theories on either side of it. Both word and island-collision events are assigned priority scores derived from the quality scores of the words that they contain and are inserted into the event queue according to their priorities.

(3) Continue selecting the top priority event from the event queue (step 2) until a theory is discovered that spans the entire utterance and is syntactically, semantically, and pragmatically acceptable as a complete sentence.

The main topic in this paper is the assignment of priority scores to the events in the above algorithm in order to guarantee that the first complete theory found will be the best scoring one that can be found. Using the quality scores assigned by the Lexical Retrieval component directly as priority scores does not ordinarily provide such a guarantee. That is, a straightforward "best-first" search strategy does not guarantee discovery of the best overall hypothesis.

Note: Although the basic island-driven strategies are presented here as involving an initial scan of the entire utterance before beginning the processing of events, there is nothing to prevent an implementation from dovetailing this initial scan with the event processing so that, for example, event processing on the early portions of an utterance could begin before the entire utterance had been heard.

3. THE SHORTFALL SCORING METHOD

3.a Assumptions

The shortfall method assumes that the quality scores assigned to word matches by the Lexical Retrieval component are additive, so that theories are appropriately assigned scores that are the sums of the scores of the word matches contained in them (scores that are basically multiplicative can be handled by using their logarithms). It also assumes that word matches have associated beginning and ending positions that correspond to boundary positions in the input utterance. In the HWIM system, the quality scores are logarithms of estimates of the relative probabilities of the correctness of theories given the acoustic evidence.

3.b The Basic Shortfall Scoring Procedure

Assume the utterance is divided by an acoustic-phonetic processor into phonetic or subphonetic segments separated by boundaries numbered logically from the beginning of the utterance. Let t(i) be the time in milliseconds of the i-th boundary in the utterance; nsegs, the number of segments in the utterance; and seg(i) be the region of the input utterance from t(i-1) to t(i), for i from 1 to nsegs.

For a word match from position i to j with score q, we will, in some systematic way, allocate the total word score q to the segments seg(i+1) ... seg(j) covered by the word match. For this discussion, let us allocate it proportional to the durations of the segments.

For a given utterance, we will determine for each segment seg(i) the maximum score maxseg(i) that can be allocated to that segment by any word match that covers the segment.* The score for any word match from i to j will hence be bounded by the sum maxseg(i+1)+ ... +maxseg(j), and the maximum score for any complete theory will be bounded by T = the sum from 1 to nsegs of maxseg(i).

Every partial theory will consist of a sequence of contiguous word matches spanning a region from some boundary i to some boundary j. Each such theory will carry with it two scores m and q, where m is the sum of the maxseg(i) for the segments covered by the sequence and q is the sum of the word scores of the theory. We will assign each theory a priority score p = T - m + q. which can be thought of as the maximum total score T for any theory minus the shortfall from this ideal to which one is committed by choosing this particular sequence of words (i.e., p = T-(m-q)). Alternatively, p can be thought of as the estimated best possible future score consisting of the score q which has already been achieved for the region covered plus the best potential score T-m for the region not yet covered (i.e., p = q+(T-m)). Because T-m is an upper bound on the possible score that can be achieved on the region not covered, the priority scores p have the characteristic that they are non-increasing as theories grow.

---

* There are several ways to actually compute such an upper bound. The simplest involves using the best possible phoneme scores assigned by the acoustic phonetic recognizer. The tightest bound comes from accumulating the maxsegs from allocated scores of actual word matches. See Section 3.j for further discussion.

3.c Strategy

In the shortfall scoring strategy, the priority scores of the individual seed events are simply the shortfall scores of the words. A priority score for a continuation event that will be an upper bound on the priority score of any words that might result from the continuation can be computed as follows: Since the Lexical Retrieval component enumerates words in decreasing order of score, the quality score of any word that results from the continuation will be no greater than that of the last word enumerated so far. Moreover, we can derive from the lexicon a lower bound on the length of a word and from this we can deduce the shortest region of the utterance that such a word could cover, and hence the smallest possible m score that such a word could have. From these two numbers, we can bound the priority score (T-m+q) of any future word and use that as the priority score of the continuation event. (This bound is excessively conservative, and in actual practice it should be possible to derive a much tighter bound. However, this argument is sufficient to guarantee that such a bound can be computed.) A preferred alternative in this case would be to have a lexical retrieval component that enumerated words directly in increasing order of shortfall. This could be done by another instantiation of the same shortfall method, recognizing words as sequences of phonemes. The lexical retrieval component in HWIM, however, did not do this.

As new theories arise from processing events linking an existing theory with a new word match, the m and q scores of an event and the new theory that it will create are simply the respective sums of the m and q scores of the old theory and the word being added to it. Thus, after assigning an m score to a word match by summing the max numbers for the segments that it covers, the m score of any new theory that includes it can be computed by a single addition.

3.d Admissibility of the Method

Claim:

The first complete spanning theory found by the shortfall scoring method will be one of the best scoring complete theories (there could be more than one) that can be found by any strategy (i.e., the algorithm is "admissible" in the conventional terminology of heuristic search).

Proof:

At the time the first complete spanning theory has been processed, every other event on the event queue (including continuation events for finding lower scoring seeds or lower scoring words to add to the ends of islands) will already have fallen low enough in its partial score (q score) that no possible match sequence in the remaining region of the utterance can bring its total score above that of the spanning theory. Also, the presence of the continuation events in the queue makes the search process complete in the sense that any word in the vocabulary would be enumerated if the process were continued long enough. Thus there is no possible word sequence across the utterance that would not be considered by this search algorithm if it were run sufficiently far. Hence, any complete theory of the utterance will have a shortfall (m-q) at least as great as that of the first complete theory discovered. Since all spanning theories have the same maxscore m = T, it follows that the first spanning theory also has the maximum possible quality score (q) of any spanning theory.

3.e Notes

Note that the process can be continued to obtain the second best complete theory, and so on. Note also that the admissibility holds for this method whether the process is left-to-right (i.e., seeds only at the left end of the utterance) or middle-out (seeds anywhere in the utterance), and that it does not require any island collision feature.

The shortfall method works with almost any type of grammar. It makes no assumptions that the grammar is finite-state, as do Markovian strategies. In the middle-out modes, it does require the linguistic consultant to have a parser (such as the bidirectional ATN parser in the HWIM system) that can take an arbitrary island fragment in the middle of an utterance and judge whether it is a possible subsequence of an acceptable sentence. In practice, it also helps if the parser can use the grammar to predict the acceptable words and classes adjacent to an island, and if the Lexical Retrieval component can use such predictions to constrain its search (as in HWIM), but this is not essential to the formal admissibility of the algorithm.

3.f Avoiding Duplicate Theories

Note that in the middle-out, island-driven strategies there are many different ways of eventually arriving at the same theory.

For example, if we have an island w with a possible word x on the left and a possible word y on the right, then we can first form the theory (xw) and then (xwy) or we can form the theory (wy) and then derive (xwy) from that. Which of these two routes is taken will depend on the scores of the words, but it is quite possible (in fact, likely) that in the course of working toward a complete theory a strategy will arrive at the same subtheory several different times by alternate routes.

If we do not include checks for the duplication of theories, then we would often get two copies of the same theory. These would forever duplicate the same predictions and theory formations, giving rise to an exponential explosion of the search process. If we include a test each time a theory is formed to determine whether that theory has been formed previously, then we can avoid this exponential process. In fact, if each time we are about to put a word event on the event queue we check the event to see if the set of word matches that it uses is the same as that of some other event, then we can terminate this duplication before making the entry on the queue and consuming the queue space (and certainly before calling the Linguistic component to check it out and make further predictions).

The check for duplication among all the events that have been created can constitute a considerable amount of testing if done in a brute force exhaustive test. However, it can be considerably reduced by indexing events by their beginning and end points or other tricks. Moreover, if one can rely on the events being generated in the order determined by the basic shortfall strategy, then the following simple check based only on the word matches at each end of an event can be used to determine whether an event is redundant (i.e., will produce the same theory as some event already generated):

> If the new word is at the left end and has the same or smaller shortfall as the rightmost word in the theory, then this event is redundant.
>
> If the new word is at the right end and has strictly smaller shortfall than the leftmost word in the theory, then this event is redundant.

The argument for the validity of this test is as follows:

In the search space we are considering, it is possible, without a check for duplication, to derive a given theory with

words $w_1, w_2, \ldots, w_k$ in 2^{k-1} different ways - one corresponding to each of the possible binary derivation trees starting with some one of the w_i as a seed, and then successively adding words either to the right or the left end. (Proof - either w_1 or w_k was chosen last, hence there are two ways to derive a string of length k for every possible derivation of a string of length k-1. There is one possible way - i.e., as a seed - to derive a string of length 1.) Of all these derivation trees, the first one that will be found is the one that uses the w_i with the smallest shortfall as a seed, and at subsequent steps adds the better (in terms of shortfall) of the two words at either end (assume for the moment that no two of the words have exactly the same score). Hence, any derivation that attempts to add a word to one end of an island when that word has a smaller shortfall than the word at the other end of the island will be duplicating a theory that has already been derived (or at least already has an event for it on the event queue). In the case of two competing seeds with the same shortfall or words at each end of an island that have the same shortfall, we have arbitrarily picked the leftmost as the preferred one, which we will permit the algorithm to follow fully, and we block the derivation of duplicates from the other one. Thus, if we have a word being added to the left end of a theory that has the same shortfall as the word at the right end, then this event is redundant, since the preferred order will generate an equivalent event that adds the left end word first.

Thus, a very simple check between the score of the word being added to a theory and the score of the word at the other end of the theory will suffice to eliminate the formation of redundant events.

3.g Fuzzy Word Matches

The above discussion does not explicitly mention the problem of finding the same word in essentially the same place but with slightly different end points and different scores. We have observed this kind of output from the Lexical Retrieval component of HWIM and indeed find it desirable to know the degree of variation possible in the end points of a word match and the appropriate degradation in score for each. However, it is wasteful to give several different events to the Linguistic component, all of which are adding word matches to a given theory that differ only in their endpoints and scores. For this reason, we have introduced a structure that groups together multiple equivalent word matches into a single entity called a fuzzy word match (or "fuzzy" for short), which is given the score of its best member. A theory containing fuzzy word matches actually represents a class of grammatically equivalent theories and carries the score of the best one.

When an event is created to add a word match to a theory containing a fuzzy word match at that end, the score of the event must be computed using a "rectified" score that takes into account the best member of the fuzzy that is compatible with the new word (i.e., has boundaries that hook up to the new word and satisfies appropriate phonological word boundary constraints). In general, when several fuzzies are adjacent, the best compatible sequence of word matches must be chosen, and when the new word match is itself a fuzzy, the best combination of one of its members with a sequence of word matches from the theory must be taken. The event is thus given the score of the best of the grammatically equivalent, non-fuzzy events for which it stands. (Note that the score for an event is the same as the score for the theory which will result from it.)

If word matches returned by the Lexical Retrieval component are grouped into fuzzy matches whenever possible, and word events are given appropriately rectified scores, then the above admissibility result still holds (i.e., the first complete theory processed will be the best). The only difference (aside from the elimination of separate processing for grammatically equivalent theories) will be that certain word events (i.e., those whose new word(s) is (are) compatible only with a less-than-best path through the existing theory) will be formed earlier than they otherwise would have. However, these events will still be placed on the queue with the correct score (i.e., the score of the best path through the resulting theory) so that they will reach the top and be processed in exactly the same order as they would in the strategy without fuzzies.

3.h Comparison with Known Optimal Algorithms

The shortfall scoring method is similar in some respects to the well-known branch and bound technique, except for the fact that the space of possible solutions is determined by a grammar, and the characteristic in the middle-out version that the same partial interpretation may be reached by many different paths. It can also be modeled as an example of the A* algorithm of Hart, Nilsson, and Raphael (1968) for finding the shortest path through a graph, where, in this case, the nodes in the graph are partial interpretations of the utterance, and the connections in the graph correspond to the seed and word events. Consequently, it shares with that algorithm a certain kind of optimality that Hart, Nilsson, and Raphael prove - i.e., among other algorithms in its class, it explores the fewest hypotheses possible for a given bounding function while still assuring the discovery of the best hypothesis. It is simpler than the general A* algorithm, however,

in that we are looking for the best scoring node, and we are not interested in scores of paths leading to that node (in fact all such paths have the same score in our case). The simple argument given previously suffices to show the admissibility of the shortfall method, whereas the general A* algorithm is more complicated.

3.i Computing the MAXSEG Profile

Measuring the shortfall from any maxseg profile that is a per word upper bound of quality score would be sufficient to assure the theoretical admissibility of the shortfall method. However, the tightness of the upper bound affects the number of events tried and partial theories created in the search for a successful interpretation (i.e., the "breadth" of the search). By assigning the upper bound as a segment-by-segment profile determined by allocated shares of actual word match scores, a fairly tight upper bound can be achieved, which tends to minimize the breadth of search. In HWIM, the maxseg profile was computed from the word matches found so far (the best of which are found first). When occasionally a word match is found that raises the maxseg for some segment, all events are appropriately rescored.

3.j Discussion

When using the shortfall method, the overwhelming tendency is that an event adding a new word to an island will pick up additional shortfall and fall some distance down in the queue. The result is that other events are processed before any additional work is done on that island. (Occasionally, the new word is the best word in its region and buys no additional shortfall, but this is a rarity.) The distance that this new event falls down the queue is determined by the amount of additional shortfall that it has just picked up and the shortfalls of the events that are competing with it on the queue. This distance directly affects the degree of "depth-first" vs. "breadth-first" processing done by the algorithm. If the new word scores well, the event falls only slightly, and few, if any, alternate events are processed before it. In this case the algorithm is relatively depth first. If the new word scores badly, the event falls further down the queue, many more alternative events have priority over it and the algorithm is more breadth first.

The above characterization is only an intuitive approximation, since the actual number of events processed before the new event is considered depends on the number of new events that will be generated by the intervening events that will also score higher

than this one. In some cases, the number of such events can be extensive. The general effect, however, is that the shortfall scoring method provides a dynamically varying combination of depth-first and breadth-first search which is determined by the relative qualities of the events that are in competition.

Unfortunately, experience with the HWIM system has shown that the shortfall algorithm is excessively conservative. It amounts to assuming that any theory will obtain the maximum possible scores in the regions not yet covered. This is clearly overly optimistic in almost all cases, and it in fact leads to an excessively breadth first search. (For more details, see Section 7.)

4. DENSITY SCORING WITH ISLAND COLLISIONS

Density scoring is a fundamentally different priority scoring method. It uses a priority score which is the quality score of a theory divided by the duration of the region that it covers. One way to view this strategy is to consider again the task of estimating the expected score to be achieved in the region not covered by a theory and consider estimating this score as a direct extrapolation of the same score per millisecond that has already been achieved - i.e., add to the current score an estimated potential score consisting of the score density of the current theory times the duration of the region not covered by that theory. Since the resulting total estimated score is just the score density of the theory times the total duration of the utterance, and the total duration of the utterance is a constant, we can compare only the score densities of the theories themselves and achieve the same decisions.

When we think of the score density as an extrapolation of the score already achieved by a theory into the region not yet covered we are clearly no longer obtaining an upper bound on the possible future score an event might lead to. Hence, the previous proof of admissibility used for the shortfall method no longer applies. In particular, whereas T minus the shortfall is a monotonically decreasing function as an island grows, the score densities can get smaller when a bad word is picked up and then get larger again as the theory grows and picks up better words (thus averaging the score of the bad word over a larger duration). Hence, it is not true that the score density of descendants of an event must be no greater than that of the event itself.

However, when used with an island collision feature that allows one to combine together in one step the word lists of two different theories that are noticing the same word from opposite

directions, the density method also guarantees that the first complete theory found is the best one. To prove this, we must use a different argument than for the basic shortfall strategy. The argument depends on the ability to derive subparts of a theory independently from different seeds - i.e., the middle-out control strategy is essential for the admissibility of the density scoring method.

Lemma:

Using the density scoring method in a middle-out strategy with island collision events, any theory covering any region of the utterance can be derived by a sequence of events all of which have a score density no less than that of the theory itself.

Proof:

By induction on the number of words in the theory:

(1) The hypothesis is trivially true for one-word theories by means of a seed event.

(2) Suppose that the hypothesis is true for theories of k or fewer words and that we have a theory of k+1 words with density d. Assume that the theory consists of the sequence of words $w_0w_1...w_k$.

Case a. If the theory $w_1...w_k$ (i.e., all but w_0) has density not less than d, then by the inductive hypothesis it has a derivation whose events all have density not less than d, and this derivation plus the event to add w_0 will constitute the desired derivation of the complete theory.

Case b. Similarly if the theory $w_0...w_{k-1}$ has a density no less than d, there is a suitable derivation of that theory that can be extended to a derivation of the complete theory with density no less than d by adding w_k.

Case c. If neither a nor b is the case, then since $w_1...w_k$ has density less than d, therefore w_0 must have density greater than d. Let j be the smallest integer such that $w_0...w_j$ has density less than d. Such a j, smaller than k, must exist since the theory $w_0...w_{k-1}$ has density less than d. Also, j must be larger than 0 since w_0 has density greater than d. Now since the density of $w_0...w_j$ is less than d, the remaining theory $w_{j+1}...w_k$ must have density greater than d. Also, since j is the smallest such, the theory $w_0...w_{j-1}$ has density greater than or equal to d.

Since these last two theories each have length smaller than k and density no less than d, by the inductive hypothesis they each have derivations using events of density no less than d. Therefore, before any events of density less than d can reach the top of the stack, both of these theories would have been processed, and both would have noticed the word w_j from opposite sides; hence an island collision event would have been constructed for the combined theory and would have the combined density d.

Corollary:

When a spanning theory of some density has been found by the middle-out density scoring method with island collisions, any spanning theory of higher density could have been completely derived using events of higher density, and thus would have been found before the theory in question. Hence, the first complete spanning theory found will be one of the best possible interpretations.

Corollary (dual algorithm):

A dual of the above lemma shows that a density algorithm that prefers the smallest rather than the largest density will guarantee the discovery of the lowest scoring theory.

5. SHORTFALL DENSITY

The above proof of the admissibility of density scoring makes no assumptions about the scoring metric whose density is being taken other than that it be additive. Hence, the density method can be applied to either the original quality score assigned by the Lexical Retrieval component, or to the local shortfall described previously, giving rise to strategies which we refer to as quality density and shortfall density, respectively. Initial experimental comparison of the algorithms (see Sec. 7) suggests that the shortfall density method is superior to quality density, which is in turn superior to the shortfall method alone. The superiority of the density methods over the shortfall method can be accounted for by the excessive conservatism (over optimistic scoring of alternative hypotheses) of the shortfall method. The superiority of the combined shortfall density method can be attributed to an improved "focus of attention" strategy as follows:

5.a Focus of Attention by a MAXSEG Profile

A major effect of scoring the shortfall from a maxseg profile is that the score differences in different parts of the utterance are effectively leveled out, so that events in a region of the utterance where there are not very good quality words can hold their own against alternative interpretations in regions where there are high quality words. This promotes the refocusing of attention from a region where there may happen to be high quality accidental word matches to events whose word match quality may not be as great, but are the best matches in their regions. If this were not done, then many second best, third best, etc. matches in the high scoring region could be considered before any theories worked their way across the low scoring regions. Thus, an apparently satisfactory and intuitively reasonable strategy for focusing attention emerges from the same strategy that guarantees to get the best scoring theory first.

Notice that in the shortfall density method, the maxseg profile is no longer serving the role of guaranteeing admissibility that it did in the shortfall method. In this case, the admissibility is guaranteed by the nature of densities and island collisions. Rather, in this method the maxseg profile is used only to provide this leveling of effort over portions of the utterance to promote the refocusing of attention from regions where there are many good quality matches to regions where the best matching possibility may not be as great. In fact, it is no longer necessary that the maxseg profile be an upper bound (although there are undesirable effects when the shortfall density goes negative).

As long as shortfall is positive, the addition of a word with no shortfall to a hypothesis will produce a longer duration hypothesis and consequently a smaller shortfall density (which counts as a better hypothesis). Consequently, such a hypothesis will be encouraged. However, when the shortfall is negative, the addition of a new word with no shortfall will similarly produce a hypothesis with longer duration and will spread the negative shortfall over a longer period producing a density score with smaller magnitude but (since it is negative) a larger value. Consequently, such a hypothesis will be discouraged and there will be a tendency to shift attention to other negative shortfall hypotheses that are shorter. Thus, when shortfall is positive, the ordering tends to prefer hypotheses that are longer (shortfall being equal), while when the shortfall is negative, the ordering prefers hypotheses that are shorter. Hence, in a collection of hypotheses whose shortfall is negative, the search strategy will strongly favor shifting attention back to shorter hypotheses rather

than pursuing longer hypotheses. This will not affect the admissibility of the algorithm, but will exacerbate the breadth of the search in the region of negative shortfall.

6. EFFICIENCY TECHNIQUES

In addition to the basic choice of priority scoring metric used for ranking the event queue, there are several efficiency techniques that can be used to improve the performance of the island-driven strategy, frequently without loss of admissibility guarantees. Two of these are the use of "ghost" words, and the selection of a preferred direction for events from a given theory.

6.a Ghost Words

Every time a theory is given to the linguistic consultant for evaluation, proposals are made for new words on both sides of the resulting island (unless the island is already against one end of the utterance). Although events can add only one word at a time to the island, and this must be at one end or the other, eventually a word will have to be added to the other end, and that word cannot score better than the best word that was found at that end the first time. The ghost words feature consists of remembering with each event the list of words found by the Lexical Retrieval component at the other end and scoring the event using the best of the ghost words as well as the words in the event proper. The result is that bad partial interpretations tend to get bad twice as fast, since they have essentially a one-word look-ahead at the other end that comes free each time an event is processed. On the other hand, an event that has a good word match at the other end gets credit for it early, so that it gets processed sooner. The ghost words feature, thus, is an accelerator that causes extraneous events to fall faster down the event queue and allows the desired events to rise to the top faster. Experimental use of this feature has shown it to be very effective in reducing the number of events that must be processed to find the best spanning event. Its addition to the shortfall algorithm does not sacrifice admissibility. It is not theoretically admissible when added to the density methods, but it appears in practice not to sacrifice much.

6.b Choosing a Preferred Direction

Again, recall that when a theory is evaluated by the linguistic consultant, predictions are made at both ends of the island. When one of the events resulting from these predictions is later processed, adding a new word to one end of the island, the

predictions at the other end of the new island will be a subset of the predictions previously made at that end of the old island. In general, words found by this new island at that end will also have been found by the old island, and if the score of the new island is slightly worse than that of the old island (the normal situation), then the strategy will tend to revert to the old island to try events picking up a word at the other end. This leads to a rather frustrating derivation of a given theory by first enumerating a large number of different subsequences of its final word sequence.

Since any eventual spanning theory must eventually pick some word at each end of the island, one could arbitrarily pick either direction and decide to work only in that direction until the end of the utterance is encountered, and only then begin to consider events in the other direction. This would essentially eliminate the duplication described above, but could cause the algorithm to work into a region of the utterance where the correct word did not score very well without the benefit of additional syntactic support that could have been obtained by extending the island further in the other direction for a while.

Without sufficient syntactic constraint at the chosen end, there may be too many acceptable words that score fairly well for the correct poorly scoring word to occur within a reasonable distance from the top of the queue. By working on the other end, one may tighten that constraint and enable the desired word to appear (although this can never cause a better scoring word to appear than those that appeared for the shorter island).

A flag in the HWIM system causes the algorithm to pick a preferred or "chosen" direction for a given theory as the direction of the best scoring event that extends that theory, and to mark the events going in the other direction from that theory so that they can be used only for making tighter predictions for words at the chosen end. This is accomplished by blocking any events noticing one of the ghost words at the inactive end of an event if that event is going counter to the chosen direction. This blocking, alone, eliminates a significant number of redundant generations of different ways to get to the same theory. An even greater improvement is obtained by rescoring the events that are going counter to the chosen direction by using the worst ghost at the other end rather than the best ghost. Since only word matches that score worse than any of the ghosts at that end are being sought by these events, this is a much better estimate of the potential score of any spanning theories that might result from these events.

The effect of rescoring the events in the non-chosen direction using the worst ghost is that, in most cases, these events fall so low in the event queue as to be totally out of consideration. Only in those cases where there was little syntactic constraint in the chosen direction and the worst matching word at that point was still quite good, do these events stay in contention, and in those cases, the use of the worst ghost score provides an appropriate ranking of these events in the event queue.

6.c Nearly Admissible Algorithms

The heuristics of ghosts and preferred direction when added to the basic shortfall algorithm improve efficiency without affecting the formal admissibility of the algorithm. Similarly the combination of the shortfall and density algorithms does not affect the formal admissibility of the density algorithm. When adding ghosts and direction preference to the density algorithms, however, this is not necessarily the case (at least the Lemma proving the admissibility of the density method no longer goes through). It is not obvious whether these variations of the density algorithm are admissible or not. However, the basic admissibility characteristics of the algorithm remain in effect in any case, with at worst a slight chance of a non-optimal interpretation being found in pathological circumstances. We can characterize such algorithms as "nearly" admissible -- i.e., adaptations of admissible algorithms that guarantee to get the best interpretation except possibly in very low probability exceptional circumstances. Empirically, as shown below, these nearly admissible algorithms appear to have all of the advantages of the provably admissible ones (i.e., not finding incorrect interpretations) while gaining the advantages of the efficiency heuristics.

7. EMPIRICAL COMPARISON OF THE DIFFERENT STRATEGIES

In the HWIM Speech understanding system, approximations to the shortfall and density algorithms have been implemented and tested. The major approximation is that continuation events are not implemented, but instead the initial values of n and k are chosen large enough that one believes that the correct interpretation of the utterance is found before any of the continuation events would have reached the top of the queue. If such is the case, then all of the decisions made by the approximation are the same as those of the admissible theoretical algorithm, and hence the first complete theory found will still be guaranteed to be the best. There are

other approximations that are less justifiable, due to bugs and some rectifiable (but not rectified) discrepancies between the actual implementation and the theoretical algorithm. These differences are believed to be minor.

Details of the system's general performance are found in (Woods et al., 1976). Comparative performance results on a set of 10 utterances for the shortfall (S), shortfall density (SD), and quality density (QD) scoring strategies are shown in Table 1 below. The option of using the quality score (Q) alone as a priority score is given for comparison.

	Q	QD	S	SD
Correct first interpretation	4	3	0	5
Incorrect first interpretation	2	0	0	0
No response	4	7	10	5
Average number of theories processed *	49	82	100	73

Table 1. Comparison of different priority scoring functions.

These experiments were run using the ghosts, island-collision, and preferred direction heuristics with a resource limit of 100 theories to process before the system would give up with no response. The ten sentences used for the test were chosen at random from a test set of 124 recorded sentences.

Although a test set of only ten utterances is admittedly too small, I believe that the trends indicated in the figure are generally correct. Specifically, while using the quality score alone leads to a spanning interpretation in relatively few theories, it does so without much assurance of getting the best interpretation. In this case, only two-thirds of its answers are correct. All of the other methods consider more theories in an effort to make sure that the best interpretation is found. Consequently they found fewer spanning interpretations within the resource limitation but found no incorrect interpretations. We did not try running the quality scoring strategy beyond the first interpretation to see if a better interpretation could be found

* This average is computed over 9 sentences, omitting one for which the system broke due to a bug.

since, among other things, it is nontrivial to decide when to terminate such a process.* Running in this mode, one could easily enumerate more theories than the other methods and still not have any guarantee that the best interpretation had been discovered.

None of the versions of admissible algorithms found incorrect interpretations, so the reliability of their interpretations, when they get them, is 100% (providing the acoustic phonetic analysis of the input utterance does not cause some incorrect interpretation to score higher than the correct one, a situation that occurs sometimes in the HWIM system, but was not a factor in this experiment). Unfortunately, the shortfall strategy alone is so conservative in doing this that it failed to find any interpretations within the resource limit. Both of the density methods are clearly superior to the straight shortfall method. (Incidentally, the left-to-right shortfall strategy also failed to get any interpretations within the resource limit.)

The shortfall density strategy ranked superior to the quality density strategy in terms of the number of events that needed to be processed to find the first spanning interpretation and consequently found more correct interpretations within the resource limitations.

The effects of the island collision (C), ghosts (G), and preferred direction (D) heuristics are shown in Table 2 (where SD+0 means shortfall density without collisions, ghosts, or chosen direction, SD+C means shortfall density with island collisions, etc.). The inclusion of a heuristic does not always guarantee that the system will understand an utterance in fewer theories, but the pooled results shown (note especially the series SD+0, SD+G, SD+GD,

* Mostow (1977) gives a partial description of the criteria used in the Hearsay II system for making this decision, but the method is not algorithmic and is based on the assumption that any partial solution that is locally better than a found solution I (and that can be extended to a globally superior solution I") can be extended step by step into I" so that the partial solution I' at each step is locally superior to I. He makes no attempt to prove that such a sequence of partial solutions exists and appears only concerned with whether a search strategy can find one. In fact, there are situations in which no such sequence of stepwise extensions exists, as can be determined by reflecting on the proof of the admissibility of the density method and the necessity of the island collision feature for the admissibility result. One can then easily construct counterexamples to Mostow's assumption.

SD+GDC) suggest that the successively added heuristics produce improvements in both accuracy and number of theories required. (Note that our formal admissibility results have been shown only for the SD+C case. The SD+GDC case is at least nearly admissible.)

	SD+Q	SD+C	SD+G	SD+GD	SD+GDC
Correct	3	3	3	4	5
Incorrect	0	0	0	0	0
No response	7	7	7	6	5
Average number of theories processed *	83	78	81	76	69

Table 2. The effects of island collisions, ghosts, and direction preference.

8. COMPARISON WITH EXISTING SPEECH UNDERSTANDING SYSTEMS

8.a BBN HWIM

The variations on 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 deliberately inadmissible strategies, including strictly left-to-right density strategies and "hybrid" strategies that start near the left end of an utterance and work left to the end and then left-to-right across the rest of the utterance. For reasons of time and resource limitations, the final test run of the HWIM system was made using one of the inadmissible hybrid left-to-right strategies (Woods, et al., 1976). Subsequently, a much smaller experiment was run to compare various control strategies on a set of ten utterances chosen 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 nearly admissible 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:

* This average is computed over 8 sentences, omitting two for which the system broke due to bugs.

	LHSDNV	SD+GCD
Correct interpretation	6	5
Incorrect interpretation	2	0
No interpretation	2	5
Average number of theories evaluated	51	76

That is, the inadmissible left-hybrid strategy found the best (and in these cases the correct) interpretation within the resource limitation in 6 of the 10 cases, while the nearly admissible shortfall density strategy found only 5 (not necessarily a significant difference for this size sample). On the other hand, the left-hybrid 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 shortfall density strategy, on the other hand, would only claim to understand 50% of the utterances, but would misunderstand a negligible fraction.

The middle-out shortfall density algorithm in the above experiments expanded only 50% more theories (and incidentally used only 30% more cpu time) than did the left-hybrid 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, except that the middle-out 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 inadmissible 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 interpretation found would be the best one.

8.b DRAGON

The DRAGON system (Baker, 1975) is the only other speech understanding system in the ARPA project that provides a guaranteed best matching 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 at position i is a

relatively straightforward local computation, although for a grammar with n states, the number of operations for each such step 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 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., person and number agreement and semantic constraints 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 leading to a given state may have to be considered in order to find 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 an amount of time proportional to the length of the utterance, being simple to compute, and guaranteeing to obtain the optimal solution. The only difficulty (aside from estimating the necessary transition probabilities) 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

* 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.

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).

An unfortunate disadvantage of the dynamic programming algorithm is that it cannot be continued to obtain the second best interpretation. It loses this ability when it throws away all but the best path leading to each state. Hence a system like DRAGON can have no way of knowing if there are two competing interpretations with very similar scores.

8.c HARPY

The CMU HARPY system (Lowerre, 1976) is a development on the DRAGON theme which gives up the theoretical 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. This gives an algorithm that carries a number of paths in parallel (the number varying 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 can closely approximate 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 practice, HARPY's threshold is set so that it introduces negligible likelihood of missing the best interpretation, thus achieving a nearly admissible algorithm in the terminology introduced above. Like the DRAGON algorithm, it cannot be continued to produce the second best interpretation.

The HARPY system has the best demonstrated performance statistics of any of the ARPA continuous speech understanding

systems. However, it derives this performance in large part from the use of a highly constraining (and advantageously structured) finite-state grammar (see Wolf and Woods, 1980). 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", "thirty" 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). Such grammar patterns significantly reduce the number of possible "distractor" hypotheses that can compete with the correct interpretation of a test sentence, even when they are not used as test sentences themselves.

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.

---

* Later references to this grammar refer to a "dynamic" branching ratio of 30. This ratio is computed by averaging the branching ratio along the paths of the correct interpretations of utterances, whereas the branching ratio of 10 results from averaging uniformly over the grammar as a whole. As a measure of the difficulty of a grammar for speech understanding, the average over the entire grammar is more appropriate, since it measures the potential for the grammar to permit viable "distractor" hypotheses that might be confused with a correct interpretation. In the actual searching of the hypothesis space for a correct interpretation, most of the hypotheses considered will in fact be such distractor hypotheses and not partial hypotheses along the correct path.

The HARPY technique appears 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, the most important of which is the combinatorics of expanding a large habitable grammar into a finite-state network. For example, the incorporation of a single context sensitive feature (such as number agreement between subjects and verbs) into a finite-state grammar requires the doubling of the number of states in a large sub net of the grammar, the incorporation of two such features requires a quadrupling of states, and so on. In the worst case, implementing the constraint of a context free grammar that the number of "pushes" for self-embedding constituents must match the number of "pops" cannot be represented with any finite number of states, necessitating finite-state approximations tat either accept sentences that the original grammar doesn't or fail to accept some that it does. Such finite-state grammars also have difficulty dealing with dynamically changing situations such as constraints on utterances that depend on previous utterances.

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 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 be considered.

8.d 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 that can happen if there are acoustic-phonetic scoring errors such that the best scoring

interpretation is not correct or if the transition probabilities of the Markov model do not agree with the test set).

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 different speaker is significantly less.

8.e Hearsay II

The Hearsay II system (Erman et al., 1980) 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 the kind of island collisions described here. However, the control strategies with which it has been run are substantially different.

The major emphasis of the Hearsay-II work has been architectural rather than algorithmic, resulting in a general system of knowledge sources (KS's) which communicate with each other via a cross referenced structure called a "blackboard." Each KS is invoked by the satisfaction of a condition called a stimulus frame (SF) associated with the KS. When a KS is invoked, it performs actions specified in a response frame (RF) which make or change entries on the blackboard, thus triggering additional KS's until a stopping condition is realized. At that time, the best overall hypothesis yet found is taken as the interpretation of the utterance, or if not complete hypothesis has been found, a combination of partial hypotheses is chosen and the system attempts to construct a semantic interpretation from that. The blackboard of the system is divided into parametric, segment, syllable, word, word-sequence, phrase, and semantic interpretation levels.

Compared to HWIM, the Hearsay-II architecture appears to encourage a kind of "pandemonium" strategy versus HWIM's emphasis on specific components interacting in specific ways with specified

orderings of queues of events. In fact the differences between HWIM and Hearsay-II in this respect are more apparent than real. The Hearsay-II system also maintains queues of things to do, and the HWIM system does in fact maintain pointers connecting data structures at different levels. A substantial architectural difference is that the structures in HWIM are tailored to the classes of algorithms being executed rather than using a very general common data structure throughout as in Hearsay.* The major difference comes down to the former's emphasis on an architecture to support a nonspecific control strategy versus HWIM's emphasis on the discovery of effective control algorithms.

The details of the algorithms used in Hearsay-II are largely relegated to the "contents of the KS's" and have been difficult to extract from available publications. As of 1976, when this paper was first written, the best available description of the Hearsay-II algorithm was extremely sketchy, reflecting the extreme of the architectural vs algorithmic emphasis. In Hayes-Roth and Lesser (1976), even the idea of formulating an explicit control strategy was rejected as "inappropriate" (because it "destroys the data-directed nature and modularity of knowledge source activity").

Hearsay-II's scoring function for hypotheses, which its authors refer to as the "desirability" of a KS, 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. Although it is

* The latter has aesthetic appeal, but the former is more efficient, as evidenced by the historical trend in Hearsay development toward moving information out of the blackboard and into specialized data structures within the different components (see Erman et al., 1980).

not possible to tell from this description exactly what Hearsay II does, we can infer some characteristics of its behavior. First of all, 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 shortfall or density methods.

The above allusion to the "current state in the time region of the RF" refers to a function S(t) 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 algorithm (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-II'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 described as 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, this "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. 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, 1976).

Since this paper was originally written, a newer paper (Hayes-Roth and Lesser, 1977) has presented additional details of the above strategy (which they call "phrase specific") and a newer control strategy called "word specific". Among the things made

clearer in the later paper are that the duration bias discussed above is parameterized and that the current state function S(t) (and a related "implicit goal state" I(t)) participate in the overall desirability calculation as separate components of a weighted sum. By appropriate settings of parameters, one could eliminated the duration bias and any of three different terms that exploit the "state": one involving a ratio of the hypothesis's "RF validity" to the smallest state in its region (discussed above), one involving the difference between the RF validity and the state (not the same as my shortfall, however), and one proportional to the maximum I(t) in the region (I(t) is specified in the paper only to the extent that "it is only a slight oversimplification to think of I(t) as the arithmetic inverse of the current state S(t)").

The "word specific" strategy differs from the "phrase specific" strategy in several ways, one of which is that the current state function S(t) represents the highest value of any word hypothesis that is incorporated into any grammatical sequence. This makes the state function very similar to the maxseg profile at a slightly larger "grain size" (i.e., in word sized pieces rather than phoneme sized pieces). However, it is not used in the same way as the maxseg profile. Both terms that use the S(t) function in the desirability computation are measures of how much a hypothesis is better than the worst value of S(t) in its region. (The maxseg profile is used to measure how much a hypothesis is worse than an estimated best covering of its region.) The word specific strategy also drops the duration biasing from several components of the desirability computation, but still retains (and increases) the duration bias in the component which their tuning parameters give the most importance. The values of the tuning parameters are also changed in the word specific case, and the paper is somewhat ambiguous about what "value" is actually used to construct the current state function in this case.

There are sufficient omissions and ambiguities even in the later paper that it is still difficult to tell how the overall control strategy actually works. One can determine, however, that the word specific strategy is somewhat more similar to the density method than the phrase specific strategy is, although it is still substantially different. Given their description, it would be possible to set the parameters of the desirability calculation to be very similar to the quality density method (although not the shortfall density method). However, depending on details of the way the desirability of KS invocations is used in the overall system, the resulting control strategy might still not be comparable to the method presented here. At any rate, they do not appear to have tried this option. The paper reports one experiment

that shows the word specific strategy to be superior to the phrase specific one, but does not discuss the effects of varying the tuning parameters to assess the relative utilities of the various components of the desirability computation. It would be nice to see a systematic study a la Paxton (see below) of the relative merits of the different options.

In summary, the emphasis of the Hearsay-II has been largely architectural and there has apparently been little success in determining the importance of the various components of their scoring functions or in uncovering the essential elements of an effective control strategy. They report that "A significant amount of tuning of the focusing parameters has been attempted. Nevertheless, the current parameter values are probably not optimal, and it seems clearly impossible to determine what the optimal values are." (Hayes-Roth and Lesser, 1977). One can speculate, given the optimality results of this paper, that the optimum parameter values may lie in a direction much closer to the density method. The relative performance of their word specific and phrase specific strategies is consistent with this conjecture. However, it is possible that some nonobvious characteristic of the Hearsay-II architecture might block their fully exploiting the density method.

8.f The SRI experiments

At SRI, Paxton (1977) 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. 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 in HWIM. 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 phrases are not combined into larger sentence fragments corresponding to HWIM's notion of islands. His events are of two types: operations to look for a word or words at a point (which he calls a "word task", comparable to our proposals to the lexical retrieval component), and events to create such predictions from a phrase (which 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, then call the Lexical Retrieval component to find all matching words, and then 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 perhaps 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.) The success of such a method would appear to depend on the ability to judge a priori, without local acoustic evidence what words were likely to appear. That is, it demands exceptionally strong syntactic/semantic predictions.

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 (i.e., rather than ranking such words according to a priori preferences assigned by the grammar). This is more similar to the way the lexical retrieval component is used in HWIM and the algorithms presented in this paper.

"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 it, 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 phase boundary, but apparently does not permit a fragment that has incomplete phrases at both ends to be assigned a priority 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) were 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 Paxton's 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 by inhibition strategy reflect what seems to have been a common experience of the various speech understanding groups in the ARPA project. Although it seemed 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 led 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 (e.g., "four" within "California") 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.

9. COST/BENEFITS OF OPTIMALITY

There is a "folk theorem" in some AI circles that admissible strategies are more expensive than approximate ones and therefore to be avoided. Our experience with various control strategies in HWIM appears to indicate that at least in the case of speech and for the island-driven shortfall density method with island collisions, the admissible method is only 30-50% less "efficient" than a straightforward "best-first" strategy and has substantial performance advantages in minimizing false interpretations.* An

* Erman et al.'s statement (Erman et al., 1980) that BBN's experiments substantiated SRI's claim that island driving was inferior to some forms of left-to-right search is incorrect.

additional characteristic of the shortfall density method with respect to efficiency is that the combinatorics of the search depend on the amount of shortfall and not directly on the length of the input. Thus as the quality of the acoustic phonetic components improve, the combinatorics of the shortfall density algorithm improve dramatically.

One might be tempted to take the performance comparisons of the HWIM system versus the Hearsay-II system (Lea, 1980) as evidence of the superiority of approximate strategies over admissible ones. 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."* Hearsay-II can in principle explore all the alternative hypotheses that the quality density strategy would and should 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.

When speaking of nonadmissible strategies, one should be careful to distinguish between arbitrary, ad hoc strategies and what I have called "nearly admissible strategies." The latter can often have all the advantages of both. In further support of the advantages of admissibility, or at least near admissibility, over

---

Their statement is apparently based on the fact that HWIM's final performance run was made using a left-to-right, nonadmissible strategy (which we believed at the time to be expedient). (See also the discussion of Paxton's result in 8.f above.) Incidentally, their statement that the HWIM system had an explicit control strategy is also incorrect. HWIM had an explicit interconnection of components, but many different control strategies were explored within this basic architecture.

* The best reported performance results of the Hearsay II system are based on the same highly constrained, branching ratio 10 grammar used by HARPY (see section 8.c above). HWIM, on the other hand, used a general ATN grammar with estimated average branching ratio 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." Another factor affecting the relative performance of HWIM versus both HARPY and Hearsay-II is that the latter take their dictionary pronunciations from averaged actual speech of specific speakers. HWIM attempts the more difficult task of synthesizing them by rule from phonetic pronunciations from a pronouncing dictionary of American English. HWIM consequently neither requires nor uses any speaker training.

ad hoc search strategies, I should point out that the performance of the HARPY system was consistently superior both in speed and accuracy to that of Hearsay-II on the same grammar and vocabulary and with the same acoustic front end. (This is not entirely fair since Hearsay-II carried a lot of architectural baggage and was not as finely tuned as HARPY. However, it is clearly not a victory for the ad hoc approach.) Although the HARPY developers make much of the fact that their "beam search" technique gives up the guarantee of admissibility for efficiency, the HARPY algorithm in fact owes much of its success to being a nearly admissible algorithm, derived as discussed above from an admissible dynamic programming algorithm.

I would argue therefore that it is premature to rule out admissible algorithms as undesirable or inappropriate for speech understanding. In fact, preliminary evidence suggests that admissible algorithms or at least "nearly admissible" algorithms are to be preferred.

10. CONCLUSIONS

We have presented two basic priority scoring methods, shortfall and density scoring, that provide admissible search strategies for finding the best matching interpretation of a continuous speech utterance, with no limitations to finite-state grammars and without exhaustively enumerating all possible interpretations. Moreover, the two methods can be used in conjunction, and the combined method appears to be more efficient than either of the methods by themselves. We have also presented several heuristics that can be used with these basic strategies to produce admissible or nearly admissible algorithms that appear to have all of the advantages of the provably admissible ones while exploring fewer hypotheses. Although the methods are presented here in the context of speech understanding systems, analogous methods are applicable to other perceptual tasks such as vision, with appropriate generalizations of segment, word, and phrase.

The density scoring method is especially interesting, since it is not an instance of the "optimal" A* algorithm and (at least for the speech understanding problem) appears to be superior to the corresponding A* algorithm (the shortfall method) in the number of hypotheses that need to be explored to obtain the best matching solution. It apparently gains this superiority from its ability to work on different parts of the solution independently and combine them by the mechanism of island collision. This is similar in some respects to the use of lemmas in a theorem proving system. The

density method is not applicable to as wide a class of problems as the general A* algorithm, but should be applicable to any "covering" problem where scores are accumulated from partial hypotheses that can be said to "cover" some analog of a region.

Acknowledgment

This research was supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by ONR under Contract No. N00014-75-0533.

References

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, IEEE 78CH1285-6 ASSP, Tulsa, OK, April, 1978.

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

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

Erman, L.D., Hayes-Roth, F, Lesser, V.R., and Reddy, D.R. (1980)
"The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty," Computing Surveys 12 (2), June, 1980, pp. 213-253.

Hart, P., Nilsson, N., and Raphael, B. (1968)
"A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Trans. Sys. Sci. Cybernetics, July, Vol. SSC-4, No. 2, pp. 100-107.

Hayes-Roth, F, and Lesser, V.R. (1976)
"Focus of Attention in a Distributed-Logic Speech Understanding System," Conference Record, IEEE Int'l Conf. on

Acoustics, Speech and Signal Processing, ICASSP-76, Philadelphia, Pa., April, 1976.

Hayes-Roth, F. and Lesser, V.R. (1977)
"Focus of Attention in the Hearsay-II Speech Understanding System," in Proc. 5th Int'l Joint Conf. on Artificial Intelligence, Cambridge, Mass., 1977, pp. 27-35.

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

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

Mostow, D.J. (1977)
"A Halting Condition and Related Pruning Heuristic for Combinatorial Search," in CMU Computer Science Speech Group, Summary of the CMU Five-year ARPA Effort in Speech Understanding Research, Carnegie-Mellon University, Computer Science Department, Technical Report, 1977.

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

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

Wolf, J.J. and Woods, W.A. (1980)
"The HWIM Speech Understanding System" in Wayne Lea (Ed.) Trends in Speech Recognition, Prentice-Hall, Engelwood Cliffs, N.J.

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. (1978)
"Theory Formation and Control in a Speech Understanding System with Extrapolations Toward Vision," in A.R. Hanson and

E.M. Riseman (Eds.) Computer Vision Systems, Academic Press, New York, pp. 379-390.

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