Layering Predictions: Flexible use of Dialog Expectation in Speech Recognition

Sheryl R. Young, Wayne H. Ward and Alexander G. Hauptmann

Carnegie Mellon University
School of Computer Science
Pittsburgh, PA 15213

Abstract[1]

When computer speech recognition is used for problem solving or any plan based task, predictable features of the user's behavior may be inferred and used to aid the recognition of the speech input. The MINDS system generates expectations of what will be said next and uses them to assist speech recognition. Since a user does not always conform to system expectations, MINDS handles violated expectations. We use pragmatic knowledge to dynamically derive constraints about what the user is likely to say next. Then we loosen the constraints in a principled manner to generate layered sets of predictions which range from very specific to very general. To enable the speech system to give priority to recognizing what a user is most likely to say, each prediction set dynamically generates a grammar which is used by the speech recognizer. A different set of grammars is created after each user utterance. The grammars are tried in order of most specific first, until an acceptable parse is found. This allows optimal performance when users behave predictably, and displays graceful degradation when they do not

1. Overview

One of the biggest problems in computer speech recognition is coping with large search spaces. As search space size decreases, recognition performance increases. Most current speech recognizers use some form of syntactic and semantic knowledge to constrain the search space. Only utterances that are semantically and syntactically acceptable are searched for in the input speech signal. In a structured task such as problem solving, additional pragmatic knowledge sources are available for constraining search spaces. In these applications, predictive use of constraints derived from problem solving plans and context have been shown to significantly reduce search space and improve recognition [28,27,11]. To allow for diverse and unconventional user behavior, we need a principled manner for relaxing contextual constraints when they are violated. In order to do this, we organize constraints into sets that are successively more restrictive, called "layers". When some constraints are violated, we use the non-violated constraints from a less restrictive layer to reduce search space. Additionally, the flexible use of predictions allows us to derive constraints from less reliable knowledge sources. Users that behave consistent with the dynamically derived expectations can benefit greatly from enhanced recognition and the system will show a graceful degradation for those who do not. Finally, the flexible application of constraints derived from pragmatic knowledge sources will become increasingly important as we move toward larger domains and spontaneous speech recognition where search space size increases exponentially.

In this manuscript we describe an approach for using layered contextual constraints to dynamically circumscribe the search space for words in a speech signal. The implemented system (MINDS) uses these layered constraints to guide the search for words in our speech recognizer. Hence, dialog and plan based constraints are used to actually guide the initial recognition process as opposed to correct possible recognition errors or select among candidate word alternatives. For our recognizer, we use a modified version of the SPHINX [15]large vocabulary, speaker independent, continuous speech recognition system. We describe our algorithm for creating layered predictions and report the results of an experiment to evaluate search space reduction and recognition performance improvements.

2. Prior Use of Constraints in Speech Recognition

Large search space is a pressing problem in speech recognition. Essentially, speech recognition systems attempt to recognize phonemes or words in a digitized speech signal. Phoneme recognizers must also store patterns for combining the phonemes into words. There are many reasons this is difficult. The same phoneme can be compressed or expanded in time. Its pattern is influenced by the surrounding phonemic context. Words themselves have alternate pronunciations. When one considers connected speech, the above problems are further compounded by problems in identifying word boundaries

---

[1] This research was sponsored by the Defense Advance Research Projects Agency (DOD), ARPA Order No. 4976, monitored by the Air Force Avionics laboratory under contract F33615-84-K-1520.

and difficulties dealing with word juncturing effects. Word pronunciations are influenced by context and frequently phonemes are combined and omitted. Speaker independent speech recognition further compounds the pattern recognition problem because different speakers display different accents, different rates of speech and have different vocal tract characteristics. Thus, the signal patterns of different words are occasionally more alike than the patterns associated with a single word pronounced by different people. As this search space size increases, the ambiguity of the signal becomes greater and the recognition accuracy decreases.

Knowledge can be used to constrain the exponential growth of a search space and hence increase processing speed and recognition accuracy [19,6,13]. For example, the HARPY system [19] achieved a six order of magnitude search space reduction by using a finite state grammar which ensured that no syntactically or semantically inaccurate word sequences were matched in the speech signal.

Currently, the most common approach to constraining search space is to use a language model or grammar. The language models or grammars used for speech recognition dictate legal word sequences. Normally they are used in a strict left to right fashion and embody syntactic and semantic constraints on individual sentences. These constraints are represented in some form of probabilistic or semantic network which does not change from utterance to utterance [14,3,13]. A variant on this approach is to emphasize semantic structure over syntactic constraints, as exemplified in case frame parsing of speech [12].

The work on caseframe parsing provides two important insights into the speech recognition problem. First, the work demonstrates that the semantic information contained in a caseframe alone does not give enough constraint to significantly improve speech. Like other systems emphasizing semantic structure over syntactic constraint [9] this system leaves too much ambiguity in the syntactic combination possibilities and consequently shows poor recognition results. Secondly, the work demonstrates the need to apply constraints earlier in the recognition process than at the parsing level. These system generate word hypotheses (usually in the form of a word lattice) and then use semantic and syntactic constraints to select meaningful sequences of words from the set of all hypothesized words. The bottom up processing of the speech signal input results in the generation of too many word hypotheses for effective parsing. In general, those systems applying knowledge early in the recognition process (to guide the word search) have shown better results.

There are many other knowledge sources in addition to syntax and semantics. Typically, these are clustered into the category of pragmatic knowledge. Pragmatic knowledge includes inferring discourse and domain plans, using context across clausal and sentence boundaries, determining local and global constraints on utterances, and dealing with definite and pronominal reference. Work in the natural language community has shown that pragmatic knowledge sources are important for understanding language. People communicate to accomplish goals [22], and the structure of the plans to accomplish them are well understood [20,21,7]. Goals, plans and context are important for understanding implied information, understanding unexpected information and for providing helpful responses [24,25, 5,1,18,10] [4].

In the past, dialog level knowledge sources were used in speech to either correct speech recognition errors [8,2] or to disambiguate spoken input and perform inferences required for understanding [17,16,14, 23,26]. In these systems, pragmatic knowledge was applied to the output of the recognizer.

More recently, pragmatic knowledge sources have been used predictively to circumscribe the search space for words in a speech signal [28,11,27]. The MINDS system used an elaborate dialog model to infer plans, perform both domain plan tracking and discourse plan tracking [18], deal with clarification subdialogs and dynamically compute constraints using local and global focus. These knowledge sources were used predictively to dynamically restrict the sequences of words which could be matched in a speech signal. However, the first version of the MINDS system required users to act according to one of the plans known to the system. This worked well as long as the user conformed. There was no strategy for detecting and recovering from violations of these plan based constraints. We have made the system more robust with respect to these violations by introducing the mechanism of "layered predictions".

In the next section we describe the use of plans to limit search space and the algorithms which enable the MINDS system to use a layered set of predictions to dynamically modify the search space for words.

3. Plan Based Constraints

The idea underlying the MINDS system is that tracking all information communicated (user questions and database answers) enables a system to infer a set of user goals and possible problem solving plans for accomplishing these goals. Here we refer to two types of plans, those for performing problem solving within a domain and those that describe what a user may do next, or discourse plans. Thus, we can deal with both user clarifications and subdialogs. Additionally, dialog history enables a system to track goal and plan modifications. In the convention of Newell and Simon (1972) and ABSTRIPS, these goals and domain plans are represented hierarchically. *For example, in the domain of dealing with disabled ships, a goal state would be finding a replacement ship.* Additionally, like MOLGEN, constraints can be propagated from earlier information to many layers of the hierarchy. Thus, nodes in the domain plan hierarchy are only partially expanded to allow for modification and restrictions which are encountered during the dialog interaction. As each new input sentence is spoken, the system analyzes the utterance to determine the concepts expressed and uses these concepts to activate goal states and identify discourse plans. To derive constraints on future utterances, active goal states and the set of discourse plans are assessed to determine legal next states. *For example, when finding a replacement ship, some of the legal next states which follow a question about the ships in some region are questions about more ships in the region, subdialogs to clarify the last query or database answer,* *questions about availability of these ships, and questions about the ships[1] equipment.* Because speech systems use grammars to guide word transitions, we associated a list of requited and optional concepts with each goal state *(e.g. concepts associated with a goal state for ship equipment include equipment, weapons, aircraft, electronics, etc.).* The list of possible next states is used to generate a set of possible concepts which could be spoken in the next utterance. This set is then limited by local and global focus which takes into account prior context, rules about reference, etc. The speech recognizer then searches only for surface forms expressing concepts in this set.

3.1. Layered Predictions

The constraints described above were quite effective in reducing search space and improving speech recognition performance (Young, Hauptmann and Ward, 1988; Hauptmann, Young and Ward, 1988). However, there were two problems with the approach:

1. The system could not deal with unexpected behavior
2. As domain size increased, the number of possible goals and plans increased and the constraint decreased.

To deal with these shortcomings, we instituted two system modifications:

1. we enabled the system to loosen its constraints by generating layered predictions
2. we incorporated additional, less reliable knowledge sources into the predictions mechanism

To overcome the problem of having multiple active goals and domain plans, we instituted three procedures. First, since it is not possible to uniquely determine which goal state has been "activated" from an incoming utterance, we designed an algorithm to select "the most probable" goal state. Here we preferred goal states that were both complete and most likely to follow given the previous goal states activated, particularly those that were consistent with active goals at higher levels of the hierarchy or those which were clarifications of the last question posed by the user. Secondly, we maintained a list of all incomplete active goals, including those which were not hierarchically embedded. These active goals were used to generate some alternate predictions about what the user could say. Finally, we used all of our domain knowledge to generate a set of layered predictions about the content of the following utterance. The predictions ranged from very specific to very general. These layered predictions were rank ordered to reflect both amount of constraint as well as what the system felt the user would most likely say. It should be noted that the least constraining prediction layer allowed all domain concepts. This means that the system could cope with any statement the user might say even if its not included in the system grammar[2].

By layering predictions, we allow the system to reparse a speech signal with a different grammar until such time as a good parse is received. The ability to reparse an utterance also enables us to use less reliable knowledge sources to further constrain our predictions. Hence, we added two additional knowledge sources to the system: user domain expertise models and preference ordcrings for conjunctive goals.

Observing that system users with significant domain expertise solved problems using very different plans than novice users, we attempted to model the effects of expertise by constructing domain knowledge models of novice, intermediate and expert system users. Our user models were represented as subsets of the domain knowledge base. The models differed primarily by the existence of relations between domain objects. *For example, an expert user would know that each class of ships has a set of default equipment and is suited for particular types of tasks, while a novice user might not be aware that shiptypes are divided into ship classes.* The user models were then used to construct control schemas which specified which goal states were exclusive. *To further the last example, a control schema for an expert user would show that if the user asked about a shipclass they would not ask about default equipment.* These models were hand coded from the training set data.

Similarly, we used the training data to derive probabilistic orderings on conjunctive subgoals. These orderings told us which conjunctive goals would be executed first, second, etc. The orderings were computed across individuals (although our training data only came from two people). However, there is no reason why these could not be automatically obtained for individual system users in future systems.

Thus, the MINDS system used the following knowledge sources to derive predictions about the content of a user's next utterance:

1. knowledge of problem solving plans and goals represented hierarchically,
2. a finite set of discourse plans which indicate what the user may do next, such as clarify a prior utterance or response, return to a previously aborted topic, continue pursuing the current domain plan, etc.
3. semantic knowledge about the application domain's objects, attributes and their interrelations (a domain knowledge base),
4. domain independent knowledge about methods of speaking, appropriateness of references and partial utterances (local and global focus)
5. dialog history knowledge about information previously communicated,
6. discrete models of user domain expertise as described above, and
7. information about user preferences for ordering conjunctive subgoals

These knowledge sources were used by the prediction module to perform iterative analyses of the dialog after each input/database response pair and generate sets of restrictions on the next utterance. The predictions generated are layered. Each successive layer is less constraining than the

---

[2] However, the system cannot cope with words which are not included in the system lexicon

prior layer. The most constraining prediction set is generated using all knowledge sources listed above. The next set does not use user models and uses a larger non-overlapping set of goal states. Further sets are generated by moving upward in the goal hierarchy, allowing more goals and plans to be executed. The prediction sets become successively more general, hence the term "layered". Ultimately, the entire system grammar will be used. If this fails, an "allword" recognition is attempted where any word sequences are allowed (providing of course that the words are in the system lexicon). As described before, the predictions are used to restrict the surface forms searched for by the recognizer. The system uses these layers by first trying to parse at the most constraining layer. Should the parse fail to exceed a predetermined goodness score, the system reanalyzes the utterance using the next layer of predictions.

4. Derivation of Predictions

Each goal state is represented as a schema, which can be seen in Figure 1. The **concepts-required** and optional-concepts slot values are used to specify the concepts relevant when a user transits to the goal state. The number of concepts per goal state and the number of goal states a user could progress to next determine the size of the lexicon the speech recognition system must analyze. The control slot contains a pointer to a control schema wherever the child slot is not empty.

Figure 1: Example Goal State Schema

```
[ Shipclass
 :Concepts-Required ((Shipclass single-use
                        Child-restrictions*
                           (knoxclass perryclass)))
            Concept Times-used Restriction-pointers
 :Optional-Concepts ((Region single-use
                        Child-restrictions*
                           (Persian-gulf)))
 :Optional            (Not for expert-user)
                     True/Nil/User-consideration
 :Next-states        (Find-Replacement) goal-state
 :Parent             (Find-Replacement)
 :Children           (none)
 :Control            (none)
]
* = Computed by local and global focus
```

Figure 2: Example Control Schema

```
[Control00030 - for Find-replacement
 :Exclusive ((Shipclass    Equipment))
 :Omit       (Shiptype)
 :Order      ((.90 Shipclass .10 Mission-Info)
              (.90 Mission-Info  .10 Shipclass))
]
```

Control schemas predict whether any child states are likely to be omitted and any preferred orderings on the states for a specific system user. They are used to generate the most constraining prediction layers.

As seen in Figure 2, there are three slots in a control schema. The order slot stores information about preferred orderings among non-optional, conjunctive subgoal states. The exclusive slot stores pairs of goal states which are exclusive because the information in the first allows the user to infer the information in the second. The omit slot store a list of goal states the user omits because they are unaware of the domain concepts.

Control schemas are attached to parent goal states to predict which child states will be visited. Hence, they are also used to dynamically compute the value of the optional slot for each child schema. When a state is predicted to be omitted, the optional slot value becomes true for that cycle of input and database response. The optional and concepts-required slots are important for determining when a goal state is complete.

These structures are used to derive predictions by the following process. When an incoming utterance and database response are processed, we select the most likely set of goals being executed. If a goal is not complete, then our most constraining prediction set reflects the assumption that the user will complete the goal or clarify the last response or question. The other prediction layers do not change. If one or more goals are complete, we identify the next goal states to which a system user could transit. Identifying possible next states is the basis for each layer of predictions. This identification involves assessing both the active domain plans as well as providing for possible clarifications. Because discourse plans can themselves become active plans, we only need to provide for clarifications to incorporate both discourse and domain plans into the same layered prediction generation mechanism. Next, we take all of this possible next plan steps and goals which would follow from the just completed goal and store them. Following this, we apply our less reliable knowledge sources to further prune the set of next, most likely steps. To do this, we first use any and all knowledge of user ordering preferences and states which could be omitted. Then we back off first on the ordering information and then on the states which could be omitted. Then, since all goals and possible problem solving plans are represented in a hierarchical manner, we progressively move up a layer in the hierarchy of active goals to determine the state to which the user could next progress.[3] Once we have determined the next states, we can take the concepts associated with the states and compute restrictions on their expansions, restrictions on references given the state and the context, and restrictions on partial utterances. A more precise summary of the algorithm for generating predictions once a goal node is complete is presented below.

1. Find goal's next state (a goal node, usually a parent state), say G_i
2. Assess G_i to determine which of its subgoals (H-) are incomplete and store the incomplete suogoals in a list (L_{kl})
3. Assess G_is control schema to determine if any of the H_j's in list L_{kl} are to be omitted or are mutually exclusive with already complete subgoals. If so, create list L_{k2}, eliminating the

---

[3] The algorithm is somewhat simplified for purposes of this discussion. A forthcoming paper will discuss predictions re: clarification subdialogs and multiple non hierarchically embedded active goals

appropriate H_j's and reformulate L_{k1} so that it does not intersect with L_{k2}.

4. If there is more that one subgoal in the remaining list (either L_{k2} if it exists, else L_{k1}) create list L_{k3} by determining which of the H_j's is predicted to occur prior to the others. Eliminate this H_j from the appropriate list.
5. For each list, beginning with L_{k3} if it exists, collect all concepts associated with the state and compute any restrictions on these concepts. Then find the associated grammar networks associated with these concepts and generate a layer of predictions.
6. Find G_i's parent state and its H_j's, exclusive of G_i and repeat step 5.
7. Repeat step 6 until the tree's root node is found.

5. Results

The above described use of plans in speech recognition is currently embodied in the MINDS multimedia interactive dialog system. Users can speak, type or point to input information and both the system and the user can initiate clarification dialogs when appropriate. It uses an adapted version of the SPHINX [15] speech recognition system with a 1000 word vocabulary. Its task domain is naval resource management. Here users must query a relational database to determine whether a disabled vessel should be replaced with another vessel, scheduled for a later repair, or whether the mission should be delayed.

To test the ability of our layered predictions to both reduce search space and to improve speech recognition performance, we used an independent test set. This means that the utterances processed by the system were not used to train either the speech recognition system or the dialog model, user models or ordering preferences. Furthermore, the test set did not include any clarification dialogs.

5.1. Test and Training Sets

Our test data (10 scenarios) were adapted versions of three problem solving sessions taken from the TONE database. The TONE database is a set of speech transcripts from Naval personnel solving problems about what to do with a disabled vessel. The personnel must determine whether to delay a mission, find a replacement vessel or schedule a repair for a later date. They use a database to find necessary problem solving information. To assess improvement in speech recognition performance, we used only the three independent test scenarios adapted from the TONE database. To assess reduction in search space we used both the 3 adapted TONE scenarios in addition to seven more dialogs created by paraphrasing the original three. These scenarios were not used to train upon.

Our training data were five different problem solving scenarios from the TONE database. The training scenarios were used for writing grammars and developing user models. Domain goals and problem solving plans were derived from an abstract description of the stages and options available to a problem solver. The abstract plan descriptions were provided by the Navy.

As mentioned above, both our test and training scenarios were adapted from the TONE database. This was necessary because our database was different from the one used in gathering the TONE transcripts. While it contained the same fields, the information about particular ships differed across the two databases. To enable testing we performed the following minimal adaptations:

- Shipnames were changed to correspond to those in our database.
- Lexical entries not in our lexicon (such as 'employment schedule') were replaced with equivalent concepts from our lexicon (such as 'mission' and 'mission importance').
- Database inconsistencies were resolved in favor of the CMU database. For example, if in the naval database, ship X required capability Y for its mission but in the CMU database ship, X required mission capability Z, all references in a scenario to Y were replaced with references to Z.

These adaptations were necessary to evaluate the system without creating a new database and should have minimal impact on the integrity of the data.

5.2. Reduction in Search Space

To measure the constraint imposed by the knowledge sources, we use an index called perplexity. This is an information theoretic measure that is widely used in speech systems to characterize the constraint provided by a grammar. Perplexity represents the geometric mean of the number of alternative words at any point. Search space size for a given test sentence is computed by raising perplexity to the number of words in the sentence.

To measure the reduction in perplexity and search space it was necessary to collect test set perplexity measurements for each of the parsed sentences in two conditions:

- Total domain grammar alone
- Using predictions

Test set perplexity is the perplexity of the actual sentences parsed. It is different that total grammar perplexity because it takes into account only those alternatives which are legal next words given the grammar and specific test sentences.

To measure the test set perplexity of all the sentences in each of the test scenarios using the entire system grammar is relatively straight forward. However, measuring the test set perplexity of sentences which are parsed with layered predictions is not. Since prediction layers fail, we must report the perplexity of the layers which were successful. However, since some layers are non-overlapping, the number we report is the perplexity of the successful prediction layer merged with all the unsuccessful layers attempted.

As seen in Table 1, test set perplexity was reduced from 279.2 to 17.8.

Reduction in Branching Factor and Search Space		
Constraints used:	**grammar**	**layered predictions**
Test Set Perplexity	279.2	17.8

S3. Recognition Performance

To evaluate the effects of using layered predictions on recognition performance we used ten speakers (8 male, 2 female) who had not been used to train the recognizer. Each speaker read 20 sentences from the adapted version of the three original test scenarios provided by the Navy. Each of these utterances was recorded. The speech recordings were then run through the SPHINX recognition system in two conditions:

- using the system grammar (all legal sentences)
- using the grammar from the successful prediction layer merged with all unsuccessful layers

The results can be seen in Table 2. As can be seen, the system performed significantly better with the predictions. Error rate decreased by a factor of five. Perhaps more important, however, is the nature of the errors. In the "layered predictions" condition, 89 percent of the insertions and deletions were the word "the". Additionally, 67 percent of the substitutions were "his" for "its". Furthermore, none of the errors in the "layered predictions" condition resulted in an incorrect database query. Because both our database and the Navy's database shared the same fields and were implemented using Informix™, we could directly assess the accuracy of the SQL database queries to Informix. Hence, semantic accuracy, defined as a correct database query, was 100% in the "layered prediction" condition.

Recognition Performance		
Constraints used:	**grammar**	**layered predictions**
Test Set Perplexity	242.4	18.3
Word Accuracy	82.1	96.5
Semantic Accuracy	85%	100%
Insertions	0.0%	0.5%
Deletions	8.5%	1.6%
Substitutions	9.4%	1.4%

5.4. Use of Correct Layers

The fail-soft ability of the system enabled by the use of layered predictions can only succeed if the speech recognizer can recognize "bad" parses. This means that the speech recognizer must not produce a conrect recognition or a "parse" from a grammar layer which does not contain a representation for the utterance actually spoken.

Out of the 200 utterances in the independent test set used to evaluate relative improvement in recognition, there were 30 utterances which could not be parsed using the most specific layer of predictions. Of these, 20 could be parsed using the second most restricting layer of predictions and 10 were parsed at the third layer. For each of these utterances we evaluated whether the recognizer would correctly reject the inappropriate grammars. Our method for evaluating the goodness of a recognition parse was to use a two layer threshold. First, we evaluated the goodness of match of the entire utterance. In 70% of the cases, this alone resulted in the rejection of the inappropriate parse and consequent reparsing with the next layer gramhiar. Should a parse exceed the threshold for an entire utterance, a second evaluation was performed on each of the individual words in the parse. Should any of the words fail to exceed the word threshold, then again the utterar.ce was rejected. The final result was that all 30 of our potential "false alarms" were correctly rejected and parsed using the grammar associated with the appropriate layer of predictions. While this result is rather astonishing, it must be remembered that our sample size was only 30. However, the semantic nature of the predictions combined with the ability to generate low perplexity grammars which result in a tremendous reduction among comfusable words is also responsible for our ability to successfully reject incorrect parses. Hence, this result should be viewed as indicative that a reasonable rejection rate is possible using layered predictions.

6. Summary

The use of layered predictions derived from pragmatic knowledge sources is a very powerful technique for improving speech recognition. Layered predictions allow the recognition system to capitalize upon pragmatic knowledge sources without impaiiing their ability to recognize less likely utterances. The more consistent the users behavior, the better the recogniton. As user behavior deviates, recognition accuracy degrades gracefully and the system is capable of recovering and generating further pragmatic predictions based upon both the users expected and less expected behavior.

References

1. Allen, J. F. and Perrault, C. R. "Analyzing Intention in Utterances". *Artificial Intelligence* 75,3 (1980), 143-178.

2. Biermann, A., Rodman R., Ballard B., Betancourt, T., Bilbro, G., Deas, H., Fineman, L., Fink, P., Gilbert, K., Gregory, D. and Heidlage, F. Interactive natural language problem solving: A pragmatic approach. Conference on Applied Natural Language Processing, 1983, pp. 180 - 191.

3. Borghesi, L. and Favareto, C. Flexible Parsing of Discretely Uttered Sentences. COLING-82, Association for Computational Linguistics, Prague, July, 1982, pp. 37 - 48.

4. Carbonell, J. G. "POLITICS: Automated Ideological Reasoning.". *Cognitive Science* 2,1 (1978), 27-51.

5. Cohen, P. R. and Perrault, C. R. "Elements of a Plan-Based Theory of Speech Acts". *Cognitive Science 3* (1979), 177-212.

6. Erman, L.D. and Lesser, V R. The Hearsay-II Speech Understanding System: A Tutorial. In Lea, W.A., Ed., *Trends in Speech Recognition,* Prentice-Hall, Englewood Cliffs, NJ, 1980, pp. 340 - 360.

7. Fikes, R. E. and Nilsson, N. J. "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving". *Artificial Intelligence 2* (1971), 189-208.

8. Fink, P. K. and Biermann, A. W. "The Correction of Ill-Formed Input Using History-Based Expectation With Applications to Speech Understanding". *Computational Linguistics 12* (1986), 13-36.

9. Gatward, R.A., Johnson, S.R. and Conolly, J.H. A Natural Language Processing System Based on Functional Grammar. Speech Input/Output; Techniques and Applications, Institute for Electrical Engineers, 1986, pp. 125 -128.

10. Grosz, B. J. and Sidner, C. L. "Attention, Intentions and the Structure of Discourse". *Computation Linguistics* 72(1986), 175-204.

11. Hauptmann, A. G., Young, S. R. and Ward, W. H. Using Dialog Level Knowledge Sources to Improve Speech Recognition. *Proceedings of the Seventh National Conference on Artificial Intelligence,* 1988.

12. Hayes, PJ., Hauptmann, A.G., Carboneli, J.G. and Tomita, M. Parsing Spoken Language: a Semantic Caseframe Approach. Proceedings of COLING-86, Association for Computational Linguistics, Bonn, Germany, August, 1986.

13. Kimball, O., Price, P., Roucos, S., Schwartz, R., Kubala, F., Chow, Y.-L., Haas, A., Krasner, M. and Makhoul, J. Recognition Performance and Grammatical Constraints. *Proceedings of the DARPA Speech Recognition Workshop,* Science Applications International Corporation Report Number SAIC-86/1546,1986, pp. 53 - 59.

14. Lea, W.A. (Ed.). *Trends in Speech Recognition.* Prentice-Hall, Englewood Cliffs, NJ, 1980.

15. Lee, K. *SPHINX: Large Vocabulary, Speaker-Independent Speech Recognition.* Ph.D. Th., Carnegie-Mellon University, 1988.

16. Levinson, S. E. and Shipley, K. L. "A Conversational-Mode Airline Information and Reservation System Using Speech Input and Output". *The Bell Systems Technical Journal 59* (1980), 119-137.

17. Levinson, S. E. and Rabiner, L. R. "A Task-Oriented Conversational Mode Speech Understanding System". *Bibliotheca Phonetica 12* (1985), 149-196.

18. Litman, D. J. and Allen, J. F. "A Plan Recognition Model for Subdialogues in Conversation". *Cognitive Science 11* (1987), 163-200.

19. Lowerre, B. and Reddy, R. The Harpy Speech Understanding System. In Lea, W.A., Ed., *Trends in Speech Recognition,* Prentice-Hall, Englewood Cliffs, NJ, 1980, pp. 340 - 360.

20. Newell, A. and Simon, H. A.. *Human Problem Solving.* New Jersey: Prentice-Hall, 1972.

21. Sacerdoti, E. D. "Planning in a Hierarchy of Abstraction Spaces". *Artificial Intelligence 5,2* (1974), 115-135.

22. Searle,J. R. Indirect Speech Acts. In *Syntax and Semantics, Volume 3: Speech Acts,* New York: Academic Press, 1975.

23. Walker, D.E. SRI Research on Speech Understanding. In Lea, W. A., Ed., *Trends in Speech Recognition,* Prentice-Hall, Englewood Cliffs, NJ, 1980, pp. 294 - 315.

24. Wilensky, R. *Understanding Goal-Based Stories.* Ph.D. Th., Yale University, Sept 1978.

25. Wilensky, R.. *Planning and Understanding.* Addison Wesley, Reading, MA, 1983.

26. Woods, W.A., Bates, M., Brown, G., Bruce, B., Cook, C, Klovstad, J., Makhoul, J., Nash-Wobber, B., Schwartz, R., Wolf, J., and Zue, V. Speech Understanding Systems - Final Technical Report. Tech. Rept. 3438, Bolt, Beranek, and Newman, Inc., Cambridge, MA, 1976.

27. Young, S.R. and Ward, W.H. Towards Habitable Systems: Use of World Knowledge to Dynamically Constrain Speech Recognition. In *The Second Symposium on Advanced Man Machine Interface Through Spoken Language,,* 1988.

28. Young, S. R., Hauptmann, A. G., Ward, W. H., Smith, E. T. and Werner, P. "High Level Knowledge Sources in Usable Speech Recognition Systems". *Communications of the ACM* 32,2(1989),.