A Relaxation Method for Understanding Spontaneous Speech Utterances[1]

Stephanie Seneff

Spoken Language Systems Group
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139 U.S.A.

ABSTRACT

This paper describes an extension to the MIT ATIS (Air Travel Information Service) system, which allows it to answer a question when a full linguistic analysis fails. This "robust" parsing capability was achieved through minor extensions of pre-existing components already in place for the full linguistic analysis component. Robust parsing is applied only after a full analysis has failed, and it involves the two stages of 1) parsing a set of phrases and clauses, and 2) gluing them together to obtain a single semantic frame encoding the full meaning of the sentence. We have assessed the degree of success of the robust parsing mechanism through a breakdown of the performance of robustly parsed vs. fully parsed sentences on the October '91 "dry-run" test set. It was clear that the robust parser allowed us to answer many more questions correctly, as over a third of the sentences were not covered by the grammar. We also report here on the performance of the system on the February '92 test sentences, and discuss some issues with regard to the evaluation methodology.

INTRODUCTION

Current approaches to the language understanding aspect of spoken language systems tend to fall into two categories. In syntax-driven formulations [1,4,10], a complete syntactic analysis is performed which attempts to account for *all* words in an utterance. While providing strong linguistic constraints to the speech recognition component and a useful structure for further linguistic analysis, such an approach can break down in the presence of unknown words, novel linguistic constructs, recognition errors, and some spontaneous speech events such as false starts. In contrast, semantic-driven approaches [2,5,9] tend to derive their understanding by spotting key words and phrases in the utterance. While this approach can potentially provide better coverage and deal with ill-formed sentences, it provides less constraint for the speech recognizer, and may not be able to adequately interpret complex linguistic constructs.

This paper describes our efforts to develop a language understanding component that combines the advantages of both of these approaches. Our strategy has been to relax the constraint that the syntactic analysis must account for all of the words in an utterance. Our current implementation is a two stage process. In the first step, our parser [7] searches for a complete linguistic analysis. Failing that, constraints of the parser are relaxed to permit the recovery of parsable phrases and clauses within the sentence. These fragments are fused together using a mechanism that closely resembles our discourse history mechanism [8]. Thus the *robust* parser is able to leverage off of existing components to a large degree.

ROBUST PARSING MECHANISM

The natural language component of the MIT ATIS system makes use of a semantic frame representation of the meaning which serves as the input for database access, spoken response generation, and history management. The frame design is flexible enough to be readily extended to other domains. Domain-dependent aspects of the system are entered mainly through table-driven mechanisms that seek certain patterns in the frame, with very little explicit programming required.

Because the semantic frame is so central to our system, we felt it was appropriate to integrate the fragments provided by partial parse analysis at the frame level. Whenever a full linguistic analysis fails, a *set* of parse trees accounting for key phrases and clauses is recovered. Each parse tree is individually converted to a semantic frame, and the set of frames are combined to form a single semantic frame encoding the meaning of the entire sentence. This frame is then ready for integration into the existing mechanisms of the back-end component.

The ability to provide partial parses was achieved by modifying the parser and the grammar in minor ways. The grammar is written as a set of context free rewrite rules with constraints, and is converted automatically to a network form, where each node in the network represents a particular category (which might be a semantic name such as *a-place* or a syntactic one such as *predicate*). In full-sentence analysis mode, only the *sentence* category is allowed to terminate, and only at the *end* of the sentence. In the relaxed mode, on the other hand, a set of categories representing important clauses and

[1]This research was supported by DARPA under Contract N00014-89-J-1332, monitored through the Office of Naval Research.

phrases are allowed to terminate, and such termination can occur anywhere in the sentence.

When operating in robust mode, the parser proceeds left-to-right, initially producing an exhaustive set of possible parses beginning at the first word of the sentence. The parse that consumes the most words is then selected[2]. The parser begins again at the first subsequent word, repeating the procedure. Whenever no parses are returned, the parser advances by one word and tries again. Eventually a set of parsed phrases are returned.

In order to combine parsed fragments, we need an inheritance mechanism that is similar in many respects to our discourse model. Since we already have the capability of responding appropriately to sentence fragments such as 'aircraft" or "first class," we surmised that the same mechanism could be utilized effectively to fuse together parsed fragments *within* a single sentence. The only important distinction between such a sentence-internal history mechanism and the existing sentence-*external* history mechanism is that nothing from the internal history can be overwritten, since answers have not yet been provided to the previous parsed fragments.

In the standard history mechanism, the presence of certain attributes in the new frame masks inheritance of certain other attributes from the history. Furthermore, whenever a value for a given attribute occurs in the current frame and also in the history frame, the value of that attribute from the history is overwritten. The sentence-internal history mechanism remembers everything, however, since none of the pieces have as yet been answered. Whenever the frames are judged to be too disjoint, the system spawns additional top-level clauses, essentially producing a compound sentence. This would be the case, for example, for the input: "I'll take flight twelve oh nine. What ground transportation is available in Denver?"

An example, shown in Figure 1, will help to explain the difference between the two history mechanisms. The sentence, "What are the meals and aircraft for flight two eighty one and also for flight two oh one," is treated by the parser as three sequential entries: "What are the meals," "aircraft for flight 281," and "flight 201." If this sequence were delivered to the sentence-external history mechanism, the last phrase would be interpreted as "aircraft for flight 201." Sentence internally, however, the result would become "meals and aircraft for flights 281 and 201." Once the sentence is fully fused, the external history is brought in, and the sentence may inherit further constraints from the dialogue context, as shown in the figure, where it picks up a source and destination.

Further examples of robust parsing on sentences spoken by actual users are shown in Figure 2. In all three cases, we believe the system produced reasonable answers to the questions. The tables are omitted due to space limitations, but the verbal response gives a clear indication of the system's interpretation.

Rejection Criterion

Because the DARPA evaluation mechanism currently penalizes systems for incorrect answers, we augmented the robust parser with a capability for detecting certain key words, such as "between," which, if not properly understood, would most likely lead to an incorrect answer. Another heuristic, most relevant when a speech recognizer is included, was to refuse to answer if an unknown flight number was detected in the sentence. We used these sentences to update the discourse context, but gave a NO ANSWER response for evaluation. In addition, when the input was judged overall to be sufficiently unreliable due to recognition errors, we used a more conservative rejection criterion that excluded answers for sentences that did not receive a full parse and were suspected to require context. We used a simple algorithm (flights with no source and destination) to distinguish this set.

EVALUATION PROCEDURE

The DARPA community has been developing an evaluation scheme over the past year and a half, based on a comparison between an answer produced by the system and a set of two "min/max" answers provided by trained annotators, specifying the minimum and maximum requirement for expected entries from the database, where the maximum table addresses the overgeneration issue. The sentences for a given dialogue are presented in order to the system being tested, and it must deal with the sentence in context to come up with an appropriate answer[3]. No partial credit is given for a "nearly correct" answer, and systems are penalized for wrong answers, so that the score is defined as the difference between percent correct and percent incorrect.

The DARPA community have also been involved in a joint effort to collect a large body of common training and testing data within the ATIS domain. The data have been collected at five sites (AT&T, BBN, MIT, CMU, and SRI) under a variety of different conditions, sometimes including a speech recognizer (SRI), sometimes allowing only verbal feedback (AT&T), sometimes typed verbatim to the site-dependent system (MIT), and sometimes including a "wizard" who could rephrase the question before submitting it to the data collection system (BBN, CMU). The sentences have been labelled according to four distinct categories: Class A (no context required), Class D1 (context-dependent on only one preceding sentence), Class D (context dependent on multiple sentences)[4], and Class X (unanswerable).

Thus far, two sets of data have been released as testing material for this "min/max" whole-dialogue evaluation. The first set, the so-called October '91 "dry-run"

[2] In a more sophisticated form, the score may take into account N-best outputs and/or parse probabilities.

[3] Sentence categories are *not* known to the system.

[4] Class D and D1 have now been merged to a single class D.

```
INPUT:    "(WHAT ARE THE MEALS) AND (AIRCRAFT FOR FLIGHT TWO EIGHTY ONE) AND ALSO FOR
          (FLIGHT TWO OH ONE)"

FRAME:    [Existential clause
                    Topic: [(aircraft meal)  for:[(flight) number:(281, 201)]]]

RESPONSE: Here are  meals for flights 281 and 201 from boston to denver.
           AIRLINE  NUMBER  FROM  TO    MEALS SERVED
           UNITED    201    BOS   DEN    BREAKFAST
           UNITED    281    BOS   DEN    DINNER
          Here are the aircraft for flights 281 and 201 from boston to denver.
           AIRLINE  NUMBER  FROM  TO   AIRCRAFT    COMPANY
           UNITED   201     BOS   DEN  DC8      MCDONNELL DOUGLAS
           UNITED   281     BOS   DEN  DC8      MCDONNELL DOUGLAS
```

Figure 1: Example sentence to illustrate robust-parsing mechanism. Parentheses in the sentence indicate parsed phrases.

```
INPUT:    "(WHAT ARE THE CHEAPEST FLIGHTS FROM BOSTON TO ATLANTA)  WITH (CONNECTING FLIGHTS IN
          PHILADELPHIA)"

RESPONSE: Here are the cheapest connecting flights from Boston to Atlanta connecting in
          Philadelphia.     <shows table>
==============================
INPUT:    "(I WOULD LIKE TO FLY FROM BOSTON TO ATLANTA AND STOP IN PHILADELPHIA) (ARE THERE
          ANY PLANES) (THAT) DO (THAT)"

RESPONSE: Here are the aircraft for the flights with stops from Boston to Atlanta connecting
          in Philadelphia.  <shows table>
==============================
INPUT:    "(HOW LONG) DOES (FLIGHT TWO EIGHTY ONE) REMAIN (IN DENVER)

RESPONSE: Here are the connection times for flight 281 from Boston to San Francisco connecting
          in Denver.        <shows table>
```

Figure 2: Three examples of robustly parsed sentences, taken from the training data. Tables are omitted in the interest of conserving space.

test set, served as a testbed to confirm that the evaluation procedure can be executed and has merit. Out of a total of 362 sentences in this set, 290 were "answerable," (Class A, D, or D1). The second set is the February '92 test set, released just prior to this meeting. The data include nearly 1000 sentences, distributed equally over contributions from all five collecting sites. A subset of 687 sentences were considered evaluable. This test set is associated with a set of "official" results for all of the participating sites, mediated through NIST.

RESULTS

We report here on the results for the two DARPA test releases, and on three different systems: (1) The MIT NL (text input) system, (2) The MIT Spoken Language System (recognizer included), and (3) The MIT-SRI system (MIT NL component operating on outputs from a recognizer developed at SRI [3]). For the October '91 NL-only experiment, we give a breakdown of performance for those sentences that required robust parsing against those that received a full parse, in order to assess how much robust parsing helped. For the February '92 test set, we provide a detailed discussion of the errors for the text-input condition. We use the MIT-SRI results in an experiment to address the question of whether it is valid to penalize systems one-to-one for incorrect answers.

October '91 Test Results

A breakdown of the results for our system on text input on the October '91 test set, with robust parsing included, is given in Figure 3. All of the columns under "robust" mode would have given a NO ANSWER response without the robust parser. Over half of the answers must be correct in order to yield a net gain in score. For the Class A and Class D1 sentences, this requirement was met with a comfortable margin. Although the Class D, robustly parsed sentences yielded a greater number of incorrect answers than correct ones, this result is misleading, because the majority of the errors were not due to failures in the robust parsing algorithm. For instance,

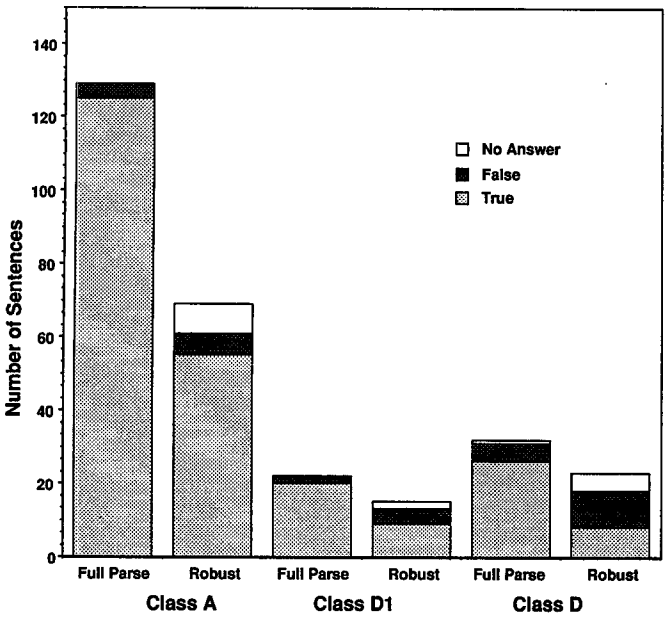

Figure 3: Results for the October '91 test set, text input, broken down by sentence type. Class A: Context Independent; Class D1: Context Dependent on a Single Query; Class D: Context Dependent on Multiple Queries. The robust parser is used only when a full parse fails.

	Correct	Incorrect	No Answer	Error
Text Input	80%	13%	7%	32.5%
MIT-SLS	61%	14%	25%	52.8%
MIT-SRI	69%	19%	12%	50.7%

Table 1: Performance results for three systems for the February '92 test set.

five sentences concerned a fare "less than one thousand dollars." A minor bug in the number interpretation routine led to an incorrect answer to all of these questions. An additional four sentences failed due to a minor problem in the external history mechanism. Overall, we were quite encouraged by the result of this evaluation, which indicates that the robust parsing mechanism provides a powerful enhancement of the system's capabilities.

February '92 Test Results

Table 1 gives performance results for the Feb '92 test set. For the text-input condition, 80% of the queries were answered correctly, and this number dropped to 61% for speech-input mode. The number of incorrect answers remained almost constant at 13%, with a corresponding large increase in unanswered questions from 7% to 25%. This is a direct result of our change in rejection strategy in going from text-input to speech-input mode. We examined in detail all of the sentences for which our text-input system produced an incorrect answer, categorizing the errors in the hopes of assessing how far away we are from the ideal goal of an error-free system.

A breakdown of the categorizations is given in Table 2. Seventeen answers fell in the category, "correct," which is to say the system produced the answer we expected it to produce, and we feel that, although the answer does not match the comparator's requirement, it is nonetheless also a reasonable answer. For instance, the answer we gave to the question, "what is the stopover," was the *location* of the stop, whereas the comparator expects, inexplicably, the *number of* stops instead. There were five essentially identical sentences asking for the number of Delta flights in differing fare classes. In all cases our count was off by one, because we included a connecting flight one of whose legs was a Delta flight. This was a consequence of a misunderstanding on our part of the rules, so we feel that the system did the right thing in this case. Two sentences were a result of the comparator refusing to accept "NIL" and a null string as the same thing. Other "correct" sentences involved an interpretation of the context, often for cases where the subject is speaking "computerese," where we think our interpretation is a valid one. Given that 20% of the errors are in this category, we believe that the comparator evaluation is probably overly rigid. It might make sense to allow some flexibility in overruling the comparator's result on a case-by-case basis.

There were 32 sentences in the category "easily fixed." It took two day's time to correct the mistakes for these sentences, although they were distributed over all aspects of the system (parse failure, meaning representation, discourse mechanism, and query generation). Some of them were clearly bugs, whereas others were simply due to incomplete understanding (such as generalizing "this afternoon" to mean "today" as well as "in the afternoon.") Six of the sentences failed due to a deficiency in our discourse mechanism specific to questions about airlines. These involved an anaphoric reference to a set of phantom flights, implied because of a preceding question about an airline. The system understood "those flights" only in the context of an existing set of flights that had been generated through a call to the database. Thus, in the sequence, "Does Delta fly between Boston and Denver," followed by "Show me *those flights*," the system was unable to understand which flights were intended. This was an interesting discourse situation, and we were happy to uncover this inadequacy in our system. Overall, while it is encouraging that it was easy to correct so many errors, it is also problematic that we continue to uncover such "minor" problems in unseen data. It is unclear how many more sets of 1000 sentences will be necessary before new bugs and inadequacies are no longer encountered.

Twenty seven sentences were judged as more difficult to correct, and their problems are about equally divided between the categories "complex meaning" and "difficult context." A particularly troublesome set for context are the sentences spoken by subjects who tend to chronically speak a staccato computerese which is difficult to distinguish from normal fragments. We are not as concerned about these sentences, because these subjects would get feedback from our system were they using it interactively,

"Correct"	Easy bugs	Hard: context	Hard: meaning	False Start	Incorrect Context	Uninteresting
17	32	12	15	3	4	4

Table 2: Breakdown of 87 errors in the MIT Text-Input February '92 test set.

which would serve to communicate to them quite clearly how the system is interpreting their staccato sentences, thus keeping the dialogue coherent. Another set of sentences that are very difficult yet probably not fruitful to correct, are "stage setting" sentences that tend to ask for too much information, such as the test-set sentence, "Please give me flight information from Denver to Pittsburgh to Atlanta and return to Denver." Our system provides a large subset of the flights requested, which is surely information overload anyway, leading the subject, in an interactive mode, to follow up with a sentence asking for information about only one leg of the trip.

The eleven remaining errors were distributed among three categories. Three were due to an incorrect analysis of a context-setting query. False starts that were deadly for the robust parser accounted for four errors. For instance, a stutter on the word "a" ("A a flight") produced the interpretation "AA" (American Airlines). Such problems are very difficult to repair, and we see no near-term solutions. An additional four errors were labelled "uninteresting," either because our system will never see such a sentence in actual operation (a request for a definition of a code like "DDEN," which is never displayed to the user by our system) or because the sentence is hopelessly obscure, such that a similar sentence would never reoccur.

The MIT-SRI System

The SRI researchers have provided us with their recognizer's outputs for three sets of data: a training data subset, the October '91 test set, and the February '92 test set. We used the training data to develop an appropriate rejection mechanism, and then we applied the results to both test sets. We decided to use the same rejection criterion for this test as for the NL-input test, without screening context dependent sentences requiring a robust parse, as we had done for the MIT recognizer inputs.

Interestingly, the error for the "MIT-SRI" system on the October '91 test set was only ten percentage points higher than that for text input, whereas the performance drop was much greater for the February '92 test set (18.2 points). We don't fully understand this difference, but apparently the recognition errors were more disruptive for the February '92 test set than for the October '91 test set. Although the SRI recognizer has a significantly better SPREC performance than the MIT recognizer (11.0% Error vs. 18.0%), our SLS system was apparently not able to take advantage of this performance improvement. The error for the MIT-only system was only 2% higher than that of the MIT-SRI system. We can think of at least two factors that may account for this surprising result. The first is that our recognizer results were obtained through filtering by TINA on 10 N-best outputs from our recognizer. If TINA could find a parsable hypothesis, then that one would be selected as the recognizer output. This meant that small errors in prepositions and the like were more likely to be corrected. The second factor is the more rigid rejection criterion used for the MIT recognizer. A larger percentage of the MIT-SRI sentences were incorrect (19% vs. 14%), and we suspect that had we used the same rejection criterion for the SRI recognizer as for the MIT recognizer the performance would have improved.

We strongly suspect that the algorithm of penalizing sentences one-to-one for incorrect answers is too steep a penalty. Because current system capabilities generally include a good discourse model as well as an ability to handle sentence fragments, it is often the case that a partially understood query provides valid information that the system can make use of in a follow-up query. For instance, if the user said: "Show me all flights from Boston to Dallas leaving Tuesday morning before ten" and the system misunderstood "Tuesday" as "Thursday," the user could simply say in a follow-up query, "On Tuesday," and the system would be able to deliver a completely correct answer. On the other hand, if the system instead refused to answer the first question (so as to maximize score), the user would have to repeat the entire sentence in order to retain the other conditions.

The only way to clearly assess whether or not systems should err in the direction of answering too much is to compare user satisfaction tests on A/B conditions. Short of this, however, it is still possible to devise an experiment to assess the degree of correctness for those answers that the recognizer misunderstood. To do this, we selected a subset of 62 utterances from the February '92 test material, representing all queries which had been correctly answered (according to the comparator) by our NL system, but incorrectly answered by the joint MIT-SRI SLS system. We have available to us a frame-based evaluation procedure that we make use of internally for comparing semantic frames generated by the recognizer against those generated from the true orthography. The scoring involves comparing a set of key/value pairs representing the set of attributes mentioned in the sentence, things like "source," "departure-time," "fare-code," "flight-number," etc. The score is computed as (correct - insertion) / (correct + substitution + deletion), where "correct" means that both the key and the value are identical between the hypothesis (NL answer)

Nsentences	Nkeys	Ncorrect	Nsub	Ndel	Nins	Score
62	213	163	22	28	21	67%

Table 3: Results of an experiment on a subset of February '92 test sentences whose orthography was correctly understood by the NL component but whose SRI recognizer outputs were incorrect. See text for further details.

and the reference (recognizer answer).

The result is shown in Table 3. There were on average about 3.5 attributes per sentence to be identified. The system identified correctly more than 3 out of every 4 attributes, with an insertion rate (recognizing additional false attributes) of 10%. This suggests to us that users would be better served if the system answered most of these questions than if the system simply said a canned phrase such as, "I'm sorry, I didn't understand you," requiring the user to reinstantiate even those attributes that had been correctly recognized.

CONCLUSIONS

Through examining a large body of speech material collected from a general population of naive users, we have reached the conclusion that it is not feasible to design a grammar that can always achieve a complete linguistic analysis of every input sentence. We have simultaneously become aware that a system that could recover a partial analysis would also be valuable for overcoming some recognition errors. We have described in this paper a capability to produce a partial analysis whenever a full parse fails, and have reported substantial performance improvements on test material as a direct consequence of this robust mechanism. We were able to leverage off of existing system components to a large extent, leading to a rapid development of the new robust parsing mechanism. This capability allowed the system to answer many more sentences than had previously been possible.

We have begun to explore some possibilities for making use of a set of N-best recognizer outputs, by parsing a *network* of paths generated through an intelligent join of the top-N candidates. We can use the frequency of occurrence of a word in the top-N candidates as a measure of its robustness, and then select a path through the network that maximizes the selection of linguistically meaningful phrases that recurred among the top-N sentences.

We have just begun to incorporate robust parsing into our data-collection procedure. We have collected data for 4 scenarios from each of 15 subjects, where the system was toggled between robust and non-robust modes half way through each subject's episode. Subjects were asked to solve the scenarios, all of which had a unique answer. Interestingly, subjects were able to find the correct answer in robust mode 90% of the time, v.s. only 70% in the non-robust mode. We take this as a clear indicator that robust mode is effective in real usage. For a further discussion of this experiment see [11].

ACKNOWLEDGEMENTS

I would like to thank Hy Murveit and John Butzberger of SRI for providing recognizer outputs for some training data and the October '91 and February '92 test set. Joe Polifroni helped diagnose the incorrect answers in the February '92 test results. Discussions with Jim Glass on robust parsing algorithms have been fruitful, and he has also played a major role in the network parsing experiments.

REFERENCES

[1] Bobrow, R.,Ingria, R., and Stallard, R., "Syntactic and Semantic Knowledge in the DELPHI Unification Grammar," *Proc. DARPA Speech and Natural Language Workshop*: 230–236, June 1990.

[2] Jackson, E., Appelt, D., Bear, J., Moore, R., and Podlozny, A., "A Template Matcher for Robust NL Interpretation,",*Proc. DARPA Speech and Natural Language Workshop:* 190–194, February, 1991.

[3] Murveit, H., "Speech Recognitin in SRI's Resource Management and ATIS Systems,"*Proc. DARPA Speech and Natural Language Workshop:* 94–100, February, 1991.

[4] Norton, L., Linebarger, M., Dahl, D. and Nguyen, N. "Augmented Role Filling Capabilities for Semantic Interpretation of Spoken Language," *Proc. DARPA Speech and Natural Language Workshop:* 125–133, February, 1991.

[5] Pieraccini, R., Levin, E. and Lee, C.H., "Stochastic Representation of Conceptual Structure in the ATIS Task," *Proc. DARPA Speech and Natural Language Workshop:* 121–124, February, 1991.

[6] Price, P., "Evaluation of Spoken Language Systems: The ATIS Domain," *Proc. Third Darpa Speech and Natural Language Workshop:* 91-95, June 1990.

[7] Seneff, S., "TINA: A Natural Language System for Spoken Language Applications," *J. Association for Computational Linguistics*, To Appear, March, 1992.

[8] Seneff, S., Hirschman, L., and Zue, V., "Interactive Problem Solving and Dialogue in the ATIS Domain," *Proc. Fourth Darpa Speech and Natural Language Workshop*: 354-359, February 1991.

[9] Ward, W., "The CMU Air Travel Information Service: Understanding Spontaneous Speech,", *Proc. DARPA Speech and Natural Language Workshop*: 127–129, June, 1990.

[10] Zue, V., Glass, J., Goodine, D., Leung, H., Phillips, M., Polifroni, J., and. Seneff, S., "Integration of Speech Recognition and Natural Language Processing in the MIT VOYAGER System," *Proc. ICASSP 91:* 713–716, May 1991.

[11] Polifroni, J., Hirschman, L., Seneff, S., and Zue, V. "Experiments in Evaluating Interactive Spoken Language Systems," These Proceedings.

[12] Zue, V., Glass, J., Goodine, D., Hirschman, L., Leung, H., Phillips, M., Polifroni, J., and. Seneff, S., "The MIT ATIS System: Preliminary Development, Spontaneous Speech Data Collection, and Performance Evaluation," *Proc. Eurospeech 91:* 537-540, September 1991.