# An Interactive Domain Independent Approach to Robust Dialogue Interpretation

**Carolyn Penstein Rosé**
LRDC 520, University of Pittsburgh
3939 Ohara St.,
Pittsburgh PA, 15260
rosecp@pitt.edu

**Lori S. Levin**
Carnegie Mellon University
Center for Machine Translation
Pittsburgh, PA 15213
lsl@cs.cmu.edu

## Abstract

We discuss an interactive approach to robust interpretation in a large scale speech-to-speech translation system. Where other interactive approaches to robust interpretation have depended upon domain dependent repair rules, the approach described here operates efficiently without any such hand-coded repair knowledge and yields a 37% reduction in error rate over a corpus of noisy sentences.

## 1 Introduction

In this paper we discuss ROSE, an interactive approach to robust interpretation developed in the context of the JANUS speech-to-speech translation system (Lavie et al., 1996). Previous interactive approaches to robust interpretation have either required excessive amounts of interaction (Rosé and Waibel, 1994), depended upon domain dependent repair rules (Van Noord, 1996; Danieli and Gerbino, 1995), or relied on the minimum distance parsing approach (Hipp, 1992; Smith, 1992; Lehman, 1989) which has been shown to be intractable in a large-scale system (Rosé and Lavie, 1997). In contrast, the ROSE approach operates efficiently without any hand-coded repair knowledge. An empirical evaluation demonstrates the efficacy of this domain independent approach. A further evaluation demonstrates that the ROSE approach combines easily with available domain knowledge in order to improve the quality of the interaction.

The ROSE approach is based on a model of human communication between speakers of different languages with a small shared language base. Humans who share a very small language base are able to communicate when the need arises by simplifying their speech patterns and negotiating until they manage to transmit their ideas to one another (Hatch, 1983). As the speaker is speaking, the listener "casts his net" in order to catch those fragments of speech that are comprehensible to him, which he then attempts to fit together semantically. His subsequent negotiation with the speaker builds upon this partial understanding. Similarly, ROSE repairs extragrammatical input in two phases. The first phase, Repair Hypothesis Formation, is responsible for assembling a set of hypotheses about the meaning of the ungrammatical utterance. In the second phase, Interaction with the User, the system generates a set of queries, negotiating with the speaker in order to narrow down to a single best meaning representation hypothesis.

This approach was evaluated in the context of the JANUS multi-lingual machine translation system. First, the system obtains a meaning representation for a sentence uttered in the source language. Then the resulting meaning representation structure is mapped onto a sentence in the target language using GENKIT (Tomita and Nyberg, 1988) with a sentence level generation grammar. Currently, the translation system deals with the scheduling domain where two speakers attempt to schedule a meeting together over the phone. This paper focuses on the Interaction phase. Details about the Hypothesis Formation phase are found in (Rosé, 1997).

## 2 Interactive Repair In Depth

As mentioned above, ROSE repairs extragrammatical input in two phases. The first phase, Repair Hypothesis Formation, is responsible for assembling a ranked set of ten or fewer hypotheses about the meaning of the ungrammatical utterance expressed in the source language. This phase is itself divided into two stages, Partial Parsing and Combination. The Partial Parsing stage is similar to the concept of the listener "casting his net" for comprehensible fragments of speech. A robust skipping parser (Lavie, 1995) is used to obtain an analysis for islands of the speaker's sentence. In the Combination stage, the fragments from the partial parse are assembled into a ranked set of alternative meaning representation hypotheses. A genetic programming (Koza, 1992; Koza, 1994) approach is used to search for different ways to combine the fragments in order to avoid requiring any hand-crafted repair rules. Our genetic programming approach has been shown previously to be orders of magnitude more efficient than the minimum distance parsing approach (Rosé and Lavie, 1997). In the second phase, Interaction with

the user, the system generates a set of queries, negotiating with the speaker in order to narrow down to a single best meaning representation hypothesis. Or, if it determines based on the user's responses to its queries that none of its hypotheses are acceptable, it requests a rephrase.

Inspired by (Clark and Wilkes-Gibbs, 1986; Clark and Schaefer, 1989), the goal of the Interaction Phase is to minimize collaborative effort between the system and the speaker while maintaining a high level of interpretation accuracy. It uses this principle in determining which portions of the speaker's utterance to question. Thus, it focuses its interaction on those portions of the speaker's meaning that it is particularly uncertain about. In its questioning, it attempts to display the state of the system's understanding, acknowledging information conveyed by the speaker as it becomes clear. The interaction process can be summarized as follows: The system first assesses the state of its understanding of what the speaker has said by extracting features that distinguish the top set of hypotheses from one another. It then builds upon this understanding by cycling through the following four step process: selecting a feature; generating a natural language query from this feature; updating its list of alternative hypotheses based on the user's answer; and finally updating its list of distinguishing features based on the remaining set of alternative hypotheses.

### 2.1 Extracting Distinguishing Features

In the example in Figure 1, the Hypothesis Formation phase produces three alternative hypotheses. The hypotheses are ranked using a trained evaluation function, but the hypothesis ranked first is not guaranteed to be best. In this case, the hypothesis ranked as second is the best hypothesis. The hypotheses are expressed in a frame-based feature structure representation. Above each hypothesis is the corresponding text generated by the system for the associated feature structure.

In order for the system to return the correct hypothesis, it must use interaction to narrow down the list of alternatives to the best single one. The first task of the Interaction Mechanism is to determine what the system knows about what the speaker has said and what it is not certain about. It does this by comparing the top set of repair hypotheses and extracting a set of features that distinguish them from one another. The set of distinguishing features corresponding to the example set of alternative hypotheses can be found in Figure 2.

The meaning representation's recursive structure is made up of frames with slots that can be filled either with other frames or with atomic fillers. These compositional structures can be thought of as trees, with the top level frame being the root of the tree and branches attached through slots. The features

**Sentence:** What did you say 'bout what was your schedule for the twenty sixth of May?

**Alternative Hypotheses:**

"What will be scheduled for the twenty-sixth of May"

```
((frame *schedule)
 (what ((frame *what)(wh +)))
 (when ((frame *simple-time)(day 26)(month 5))))
```

"You will schedule what for the twenty-sixth of May?"

```
((frame *schedule)
 (who ((frame *you)))
 (what ((frame *what)(wh +)))
 (when ((frame *simple-time)(day 26)(month 5))))
```

"Will schedule for the twenty-sixth of May"

```
((frame *schedule)
 (when ((frame *simple-time)(day 26)(month 5))))
```

Figure 1: **Example Alternative Hypotheses**

```
((f *schedule)(s who))
((f *you))
((f *schedule)(s what))
((f *schedule)(s what)(f *what))
((f *what))
((f +))
((f *what)(s wh)(f +))
((f *schedule)(s who)(f *you))
((f *schedule)(s what)(f *what)(s wh)(f +))
```

Figure 2: **Distinguishing Features**

used in the system to distinguish alternative meaning representation structures from one another specify paths down this tree structure. Thus, the distinguishing features that are extracted are always anchored in a frame or atomic filler, marked by an `f` in Figure 2. Within a feature, a frame may be followed by a slot, marked by an `s`. And a slot may be followed by a frame or atomic filler, and so on. These features are generated by comparing the set of feature structures returned from the Hypothesis Formation phase. No knowledge about what the features mean is needed in order to generate or use these features. Thus, the feature based approach is completely domain independent. It can be used without modification with any frame-based meaning representation.

When a feature is applied to a meaning representation structure, a value is obtained. Thus, features can be used to assign meaning representation structures to classes according to what value is obtained for each when the feature is applied. For example, the feature `((f *schedule)(s who)(f *you))`, distinguishes structures that contain the filler `*you` in the `who` slot in the `*schedule` frame from those that do not. When it is applied to structures that contain the specified frame in the specified slot, it returns `true`. When it is applied to structures that do not, it returns `false`. Thus, it groups the first and third hypotheses in one class, and the second hypothesis in another class. Because the value of a feature that ends in a frame or atomic filler can have either `true` or `false` as its value, these are called yes/no features. When a feature that ends in a slot, such as `((f *schedule)(s who))`, is applied to a feature structure, the value is the filler in the specified slot. These features are called wh-features.

Each feature is associated with a question that the system could ask the user. The purpose of the generated question is to determine what the value of the feature should be. The system can then keep those hypotheses that are consistent with that feature value and eliminate from consideration the rest. Generating a natural language question from a feature is discussed in section 2.3.

### 2.2 Selecting a Feature

Once a set of features is extracted, the system enters a loop in which it selects a feature from the list, generates a query, and then updates the list of alternative hypotheses and remaining distinguishing features based on the user's response. It attempts to ask the most informative questions first in order to limit the number of necessary questions. It uses the following four criteria in making its selection:

- **Askable:** Is it possible to ask a natural question from it?
- **Evaluatable:** Does it ask about a single repair or set of repairs that always occur together?
- **In Focus:** Does it involve information from the common ground?
- **Most Informative:** Is it likely to result in the greatest search space reduction?

First, the set of features is narrowed down to those features that represent askable questions. For example, it is not natural to ask about the filler of a particular slot in a particular frame if it is not known whether the ideal meaning representation structure contains that frame. Also, it is awkward to generate a wh-question based on a feature of length greater than two. For example, a question corresponding to `((f *how)(s what)(f *interval)(s end))` might be phrased something like "How is the time ending when?". So even-lengthed features more than two elements long are also eliminated at this stage.

The next criterion considered by the Interaction phase is evaluatability. In order for a Yes/No question to be evaluatable, it must confirm only a single repair action. Otherwise, if the user responds with "No" it cannot be determined whether the user is rejecting both repair actions or only one of them.

Next, the set of features is narrowed down to those that can easily be identified as being in focus. In order to do this, the system prefers to use features that overlap with structures that all of the alternative hypotheses have in common. Thus, the system encodes as much common ground knowledge in each question as possible. The structures that all of the alternative hypotheses share are called *non-controversial substructures*. As the negotiation continues, these tend to be structures that have been confirmed through interaction. Including these substructures has the effect of having questions tend to follow in a natural succession. It also has the other desirable effect that the system's state of understanding the speaker's sentence is indicated to the speaker.

The final piece of information used in selecting between those remaining features is the expected search reduction. The expected search reduction indicates how much the search space can be expected to be reduced once the answer to the corresponding question is obtained from the user. Equation 1 is for calculating $S_f$, the expected search reduction of feature number $f$.

$$S_f = \sum_{i=1}^{n_f} \left(\frac{l_{i,f}}{L}\right) \times (L - l_{i,f}) \qquad (1)$$

$L$ is the number of alternative hypotheses. As mention above, each feature can be used to assign the hypotheses to equivalence classes. $l_{i,f}$ is the number of alternative hypotheses in the $i$th equivalence class of feature $f$. If the value for feature $f$ associated with the class of length $l_{i,f}$ is the correct value, $l_{i,f}$ will be the new size of the search space. In this case, the actual search reduction will be the current number of hypotheses, $L$, minus the number of alternative hypotheses in the resulting set, $l_{i,f}$. Intuitively, the expected search reduction of a feature is the sum over all of a feature's equivalence classes of the percentage of hypotheses in that class times the reduction in the search space assuming the associated value for that feature is correct.

The first three criteria select a subset of the current distinguishing features which the final criterion then ranks. Note that all of these criteria can be evaluated without the system having any understanding about what the features actually mean.

Selected Feature:
((f *schedule)(s what)(f *what)(s wh)(f +))

Non-controversial Structures if Answer
to Question is Yes:
((when ((month 5)(day 26)(frame *simple-time)))
 (frame *schedule)
 (what ((wh +)(frame *what))))

Question Structure:
((when ((month 5)(day 26)(frame *simple-time)))
 (frame *schedule)
 (what ((wh +)(frame *what))))

Question Text:
Was something like WHAT WILL BE
SCHEDULED FOR THE TWENTY-SIXTH
OF MAY part of what you meant?

Figure 3: **Query Text Generation**

### 2.3 Generating Query Text

The selected feature is used to generate a query for the user. First, a skeleton structure is built from the feature, with top level frame equivalent to the frame at the root of the feature. Then the skeleton structure is filled out with the non-controversial substructures. If the question is a Yes/No question, it includes all of the substructures that would be non-controversial assuming the answer to the question is "Yes". Since information confirmed by the previous question is now considered non-controversial, the result of the previous interaction is made evident in how the current question is phrased. An example of a question generated with this process can be found in Figure 3.

If the selected feature is a wh-feature, i.e., if it is an even lengthed feature, the question is generated in the form of a wh-question. Otherwise the text is generated declaratively and the generated text is inserted into the following formula: "Was something like XXX part of what you meant?", where XXX is filled in with the generated text. The set of alternative answers based on the set of alternative hypotheses is presented to the user. For wh-questions, a final alternative, "None of these alternatives are acceptable", is made available. Again, no particular domain knowledge is necessary for the purpose of generating query text from features since the sentence level generation component from the system can be used as is.

### 2.4 Processing the User's Response

Once the user has responded with the correct value for the feature, only the alternative hypotheses that have that value for that feature are kept, and the rest

"What will be scheduled for the twenty-
sixth of May"

((what ((frame *what)(wh +)))
 (when ((frame *simple-time)(day 26)(month 5)))
 (frame *schedule))))

"You will schedule what for the twenty-
sixth of May?"

((what ((frame *what)(wh +)))
 (frame *schedule)
 (when ((frame *simple-time)(day 26)(month 5)))
 (who ((frame *you))))

Figure 4: **Remaining Hypotheses**

((f *schedule)(s who))
((f *you))
((f *schedule)(s who)(f *you))

Figure 5: **Remaining Distinguishing Features**

are eliminated. In the case of a wh-question, if the user selects "None of these alternatives are acceptable", all of the alternative hypothesized structures are eliminated and a rephrase is requested. After this step, all of the features that no longer partition the search space into equivalence classes are also eliminated. In the example, assume the answer to the generated question in Figure 3 was "Yes". Thus, the result is that two of the original three hypotheses are remaining, displayed in Figure 4, and the remaining set of features that still partition the search space can be found in Figure 5.

If one or more distinguishing features remain, the cycle begins again by selecting a feature, generating a question, and so on until the system narrows down to the final result. If the user does not answer positively to any of the system's questions by the time it runs out of distinguishing features regarding a particular sentence, the system loses confidence in its set of hypotheses and requests a rephrase.

## 3 Using Discourse Information

Though discourse processing is not essential to the ROSE approach, discourse information has been found to be useful in robust interpretation (Ramshaw, 1994; Smith, 1992). In this section we discuss how discourse information can be used for focusing the interaction between system and user on the task level rather than on the literal meaning of the user's utterance.

A plan-based discourse processor (Rosé et al., 1995) provides contextual expectations that guide the system in the manner in which it formulates

**Sentence:** What about any time but the ten to twelve slot on Tuesday the thirtieth?

**Hypothesis 1:**
"How about from ten o'clock till twelve o'clock Tuesday the thirtieth any time"

```
((frame *how)
 (when (*multiple*
        ((end ((frame *simple-time) (hour 12)))
         (start ((frame *simple-time) (hour 10)))
         (incl-excl inclusive)
         (frame *interval))
        ((frame *simple-time)
         (day 30)
         (day-of-week tuesday))
        ((specifier any) (name time)
         (frame *special-time)))))
```

**Hypothesis 2:**
"From ten o'clock till Tuesday the thirtieth twelve o'clock"

```
((frame *interval)
 (incl-excl inclusive)
 (start ((frame *simple-time) (hour 10)))
 (end (*multiple*
        ((frame *simple-time)
              (day 30)
              (day-of-week tuesday))
        ((frame *simple-time) (hour 12)))))
```

**Selected Feature:** ((f *how)(s when)(f *interval))

**Query Without discourse:** Was something like "how about from ten o'clock till twelve 'clock" part of what you meant?

**Query With discourse:** Are you suggesting that Tuesday November the thirtieth from ten a.m. till twelve a.m. is a good time to meet?

Figure 6: **Modified Question Generation**

queries to the user. By computing a structure for the dialogue, the discourse processor is able to identify the speech act performed by each sentence. Additionally, it augments temporal expressions from context. Based on this information, it computes the constraints on the speaker's schedule expressed by each sentence. Each constraint associates a status with a particular speaker's schedule for time slots within the time indicated by the temporal expression. There are seven possible statuses, including `accepted`, `suggested`, `preferred`, `neutral`, `dispreferred`, `busy`, and `rejected`.

As discussed above, the Interaction Mechanism uses features that distinguish between alternative hypotheses to divide the set of alternative repair hypotheses into classes. Each member within the same class has the same value for the associated feature. By comparing computed status and augmented temporal information for alternative repair hypotheses within the same class, it is possible to determine what common implications for the task each member or most of the members in the associated class have. Thus, it is possible to compute what implications for the task are associated with the corresponding value for the feature. By comparing this common information across classes, it is possible to determine whether the feature makes a consistent distinction on the task level. If so, it is possible to take this distinguishing information and use it for refocusing the associated question on the task level rather than on the level of the sentence's literal meaning.

In the example in Figure 6, the parser is not able to correctly process the "but", causing it to miss the fact that the speaker intended any other time besides ten to twelve rather than particularly ten to twelve. Two alternative hypotheses are constructed during the Hypothesis Formation phase. However, neither hypothesis correctly represents the meaning of the sentence. In this case, the purpose of the interaction is to indicate to the system that neither of the hypotheses are correct and that a rephrase is needed. This will be accomplished when the user answers negatively to the system's query since the user will not have responded positively to any of the system's queries regarding this sentence.

The system selects the feature `((f *how)(s when)(f *interval))` to distinguish the two hypotheses from one another. Its generated query is thus "Was something like HOW ABOUT FROM TEN OCLOCK TILL TWELVE OCLOCK part of what you meant?". The discourse processor returns a different result for each of these two representations. In particular, only the first hypothesis contains enough information for the discourse processor to compute any scheduling constraints since it contains both a temporal expression and a top level semantic frame. It would create a constraint associating the status of `suggested` with a representation for Tuesday the thirtieth from ten o'clock till twelve o'clock. The other hypothesis contains date information but no status information. Based on this difference, the system can generate a query asking whether or not the user expressed this constraint. Its query is "Are you suggesting that Tuesday, November the thirtieth from ten a.m. till twelve a.m. is a good time to meet?" The `suggested` status is associated with a template that looks like "Are you suggesting that XXX is a good time to meet?" The XXX is then filled in with the text generated from the temporal expression using the regular system generation grammar.

## 4 Evaluation

An empirical evaluation was conducted in order to determine how much improvement can be gained with limited amounts of interaction in the

| | Bad | Okay | Perfect | Total Acceptable |
|---|---|---|---|---|
| Parser | 85.0% | 12.0% | 3.0% | 15.0% |
| Top Hypothesis | 64.0% | 28.0% | 8.0% | 36.0% |
| 1 Question | 61.39% | 28.71% | 9.9% | 38.61 |
| 2 Questions | 59.41% | 28.71% | 11.88% | 40.59% |
| 3 Questions | 53.47% | 32.67% | 13.86% | 46.53% |

Figure 7: **Translation Quality As Maximum Number of Questions Increases**

domain independent ROSE approach. This evaluation is an end-to-end evaluation where a sentence expressed in the source language is parsed into a language independent meaning representation using the ROSE approach. This meaning representation is then mapped onto a sentence in the target language. In this case both the source language and the target language are English. An additional evaluation demonstrates the improvement in interaction quality that can be gained by introducing available domain information.

### 4.1 Domain Independent Repair

First the system automatically selected 100 sentences from a previously unseen corpus of 500 sentences. These 100 sentences are the first 100 sentences in the set that a parse quality heuristic similar to that described in (Lavie, 1995) indicated to be of low quality. The parse quality heuristic evaluates how much skipping was necessary in the parser in order to arrive at a partial parse and how well the parser's analysis scores statistically. It should be kept in mind, then, that this testing corpus is composed of 100 of the most difficult sentences from the original corpus.

The goal of the evaluation was to compute average performance per question asked and to compare this with the performance with using only the partial parser as well as with using only the Hypothesis Formation phase. In each case performance was measured in terms of a translation quality score assigned by an independent human judge to the generated natural language target text. Scores of Bad, Okay, and Perfect were assigned. A score of Bad indicates that the translation does not capture the original meaning of the input sentence. Okay indicates that the translation captures the meaning of the input sentence, but not in a completely natural manner. Perfect indicates both that the resulting translation captures the meaning of the original sentence and that it does so in a smooth and fluent manner.

Eight native speakers of English who had never previously used the translation system participated in this evaluation to interact with the system. For each sentence, the participants were presented with the original sentence and with three or fewer questions to answer. The parse result, the result of repair without interaction, and the result for each user after each question were recorded in order to be graded later by the independent judge mentioned above. Note that this evaluation was conducted on the nosiest portion of the corpus, not on an average set of naturally occurring utterances. While this evaluation indicates that repair without interaction yields an acceptable result in only 36% of these difficult cases, in an evaluation over the entire corpus, it was determined to return an acceptable result in 78% of the cases.

A global parameter was set such that the system never asked more than a maximum of three questions. This limitation was placed on the system in order to keep the task from becoming too tedious and time consuming for the users. It was estimated that three questions was approximately the maximum number of questions that users would be willing to answer per sentence.

The results are presented in Figure 7. Repair without interaction achieves a 25% reduction in error rate. Since the partial parser only produced sufficient chunks for building an acceptable repair hypothesis in about 26% of the cases where it did not produce an acceptable hypothesis by itself, the maximum reduction in error rate was 26%. Thus, a 25% reduction in error rate without interaction is a very positive result. Additionally, interaction increases the system's average translation quality above that of repair without interaction. With three questions, the system achieves a 37% reduction in error rate over partial parsing alone.

### 4.2 Discourse Based Interaction

In a final evaluation, the quality of questions based only on feature information was compared with that of questions focused on the task level using discourse information. The discourse processor was only able to provide sufficient information for reformulating 22% of the questions in terms of the task. The reason is that this discourse processor only provides information for reformulating questions distinguishing between meaning representations that differ in terms of status and augmented temporal information.

Four independent human judges were asked to grade pairs of questions, assigning a score between 1 and 5 for relevance and form and indicating which question they would prefer to answer. They were instructed to think of relevance in terms of how use-

ful they expected the question would be in helping a computer understand the sentence the question was intended to clarify. For form, they were instructed to evaluate how natural and smooth sounding the generated question was.

Interaction without discourse received on average 2.7 for form and 2.4 for relevance. Interaction with discourse, on the other hand, received 4.1 for form and 3.7 for relevance. Subjects preferred the discourse influenced question in 73.6% of the cases, expressed no preference in 14.8% of the cases, and preferred interaction without discourse in 11.6% of the cases. Though the discourse influenced question was not preferred universely, this evaluation supports the claim that humans prefer to receive clarifications on the task level and indicates that further exploration in using discourse information in repair, and particularly in interaction, is a promising avenue for future research.

## 5 Conclusions and Current Directions

This paper presents a domain independent, interactive approach to robust interpretation. Where other interactive approaches to robust interpretation have depended upon domain dependent repair rules, the approach described here operates efficiently without any such hand-coded repair knowledge. An empirical evaluation demonstrates that limited amounts of focused interaction allow this repair approach to achieve a 37% reduction in error rate over a corpus of noisy sentences. A further evaluation demonstrates that this domain independent approach combines easily with available domain knowledge in order to improve the quality of the interaction. Introducing discourse information yields a preferable query in 74% of the cases where discourse information applies. Interaction in the current ROSE approach is limited to confirming hypotheses about how the fragments of the partial parse can be combined and requesting rephrases. It would be interesting to generate and test hypotheses about information missing from the partial parse, perhaps using information predicted by the discourse context.

## References


H. H. Clark and E. F. Schaefer. 1989. Contributing to discourse. *Cognitive Science*, 13:259–294.

H. H. Clark and D. Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22:1–39.

M. Danieli and E. Gerbino. 1995. Metrics for evaluating dialogue strategies in a spoken language system. In *Working Notes of the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation.*

E. Hatch. 1983. Simplified input and second language acquisition. In R. Andersen, editor, *Pidginization and Creolization as Language Acquisition.* Newbury House Publishers.

D. R. Hipp. 1992. *Design and Development of Spoken Natural-Language Dialog Parsing Systems.* Ph.D. thesis, Dept. of Computer Science, Duke University.

J. Koza. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press.

J. Koza. 1994. *Genetic Programming II.* MIT Press.

A. Lavie, D. Gates, M. Gavalda, L. Mayfield, and A. Waibel L. Levin. 1996. Multi-lingual translation of spontaneously spoken language in a limited domain. In *Proceedings of COLING 96, Kopenhagen.*

A. Lavie. 1995. *A Grammar Based Robust Parser For Spontaneous Speech.* Ph.D. thesis, School of Computer Science, Carnegie Mellon University.

J. F. Lehman. 1989. *Adaptive Parsing: Self-Extending Natural Language Interfaces.* Ph.D. thesis, School of Computer Science, Carnegie Mellon University.

L. A. Ramshaw. 1994. Correcting real-world spelling errors using a model of the problem-solving context. *Computational Intelligence*, 10(2).

C. P. Rosé and A. Lavie. 1997. An efficient distribution of labor in a two stage robust interpretation process. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing.*

C. P. Rosé and A. Waibel. 1994. Recovering from parser failures: A hybrid statistical/symbolic approach. In *Proceedings of The Balancing Act: Combining Symbolic and Statistical Approaches to Language workshop at the 32nd Annual Meeting of the ACL.*

C. P. Rosé, B. Di Eugenio, L. S. Levin, and C. Van Ess-Dykema. 1995. Discourse processing of dialogues with multiple threads. In *Proceedings of the ACL.*

C. P. Rosé. 1997. *Robust Interactive Dialogue Interpretation.* Ph.D. thesis, School of Computer Science, Carnegie Mellon University.

R. Smith. 1992. *A Computational Model of Expectation-Driven Mixed-Initiative Dialog Processing.* Ph.D. thesis, CS Dept., Duke University.

M. Tomita and E. H. Nyberg. 1988. Generation kit and transformation kit version 3.2: User's manual. Technical Report CMU-CMT-88-MEMO, Carnegie Mellon University, Pittsburgh, PA.

G. Van Noord. 1996. Robust parsing with the head-corner parser. In *Proceedings of the Eight European Summer School In Logic, Language and Information, Prague, Czech Republic.*