Minimizing Cumulative Error in Discourse Context

Yan Qu[1], Barbara Di Eugenio[3], Alon Lavie[2], Lori Levin[2] and Carolyn P. Rosé[1]

[1] Computational Linguistics Program,
[2] Center for Machine Translation,
Carnegie Mellon University, Pittsburgh, PA 15213, USA
[3] Learning Research and Development Center,
University of Pittsburgh, Pittsburgh, PA 15260, USA

Abstract. Cumulative error limits the usefulness of context in applications utilizing contextual information. It is especially a problem in spontaneous speech systems where unexpected input, out-of-domain utterances and missing information are hard to fit into the standard structure of the contextual model. In this paper we discuss how our approaches to recognizing speech acts address the problem of cumulative error. We demonstrate the advantage of the proposed approaches over those that do not address the problem of cumulative error. The experiments are conducted in the context of Enthusiast, a large Spanish-to-English speech-to-speech translation system in the appointment scheduling domain [10, 5, 11].

1 The Cumulative Error Problem

To interpret natural language, it is necessary to take context into account. However, taking context into account can also generate new problems, such as those arising because of cumulative error. Cumulative error is introduced when an incorrect hypothesis is chosen and incorporated into the context, thus providing an inaccurate context from which subsequent context-based predictions are made. For example, in Enthusiast, a large Spanish-to-English speech-to-speech translation system in the appointment scheduling domain [10, 5, 11], we model the discourse context using speech acts to represent the functions of dialogue utterances. Speech act selection is strongly related to the task of determining how the current input utterance relates to the discourse context. When, for instance, a plan-based discourse processor is used to recognize speech acts, the discourse processor computes a chain of inferences for the current input utterance, and attaches it to the current plan tree. The location of the attachment determines which speech act is assigned to the input utterance. Typically an input utterance can be associated with more than one inference chain, representing different possible speech acts which could be performed by the utterance out of context. Focusing heuristics are used to rank the different inference chains and choose the one which attaches most coherently to the discourse context [3, 8]. However, since heuristics can make wrong predictions, the speech act may be misrecognized, thus making the context inaccurate for future context-based predictions.

Unexpected input, disfluencies, out of domain utterances, and missing information add to the frequency of misrecognition in spontaneous speech systems, leaving the discourse processor in an erroneous state which adversely affects the quality of contextual information for processing later information. For example, unexpected input can drastically change the standard flow of speech act sequences in a dialogue. Missing contextual information can make later utterances appear not to fit into the context.

Cumulative error can be a major problem in natural language systems using contextual information. Our previous experiments conducted in the context of the Enthusiast spontaneous speech translation system show that cumulative error can significantly reduce the usefulness of contextual information [6]. For example, we applied context-based predictions from our plan-based discourse processor to the problem of parse disambiguation. Specifically, we combined context-based predictions from the discourse processor with non-context-based predictions produced by the parser module [4] to disambiguate possibly multiple parses provided by the parser for an input utterance. We evaluated two different methods for combining context-based predictions with non-context-based predictions, namely a genetic programming approach and a neural network approach. We observed that in absence of cumulative error, context-based predictions contributed to the task of parse disambiguation. This results in an improvement of 13% with the genetic programming approach and of 2.5% with the neural net approach compared with the parser's non-context-based statistical disambiguation technique. However, cumulative error affected the contribution of contextual information. In the face of cumulative error, the performance decreased by 7.5% for the neural net approach and by 29.5% for the genetic programming approach compared to their respective performances in the absence of cumulative error, thus dragging the performance statistics of the context-based approaches below that of the parser's non-context-based statistical disambiguation technique. The adverse effects of cumulative error in context have been noted in NLP in general. For example, Church and Gale [2] state that "it is important to estimate the context carefully; we have found that poor measures of context are worse than none." However, we are not aware of this issue having been raised in the discourse processing literature.

In the next section, we describe some related work on processing spontaneous dialogues. Section 3 gives a brief description of our system. We discuss the techniques we used to reduce the cumulative error in discourse context for the task of speech act recognition in Section 4. Lastly, we evaluate the effects of the proposed approaches on reducing cumulative error.

2 Related Work

There has been much recent work on building a representation of the discourse context with a plan-based or finite state automaton-based discourse processor

[1, 9, 3, 7, 8, 5]. Of these, the Verbmobil discourse processor [7] and our Enthusiast discourse processor are designed to be used in a wide coverage, large scale, spontaneous speech system. In these systems, the design of the dialogue model, whether plan-based or a finite state machine, is grounded in a corpus study that identifies the standard dialogue act sequences. When the recognized dialogue act is inconsistent with the dialogue model, the systems can rely on a repair procedure to resolve the inconsistency as described in [7].

The Verbmobil repair model [7], however, does not address cumulative error in discourse context. In Verbmobil, every utterance, even if it is not consistent with the dialogue model, is assumed to be a legal dialogue step. The strategy for error recovery, therefore, is based on the hypothesis that the assignment of a dialogue act to a given utterance has been incorrect or rather that the utterance has multiple dialogue act interpretations. The semantic evaluation component in Verbmobil, which computes dialogue act information via the keyword spotter, only provides the most plausible dialogue act. The plan recognizer relies on information provided by the statistical module to find out whether additional interpretations are possible. If an incompatible dialogue act is encountered, the system employs the statistical module to provide an alternative dialogue act, which is most likely to come after the preceding dialogue act and which can be consistently followed by the current dialogue act, thereby gaining an admissible dialogue act sequence. Thus the system corrects context as the dialogue goes along. As we mentioned earlier, contrary to the assumption made in Verbmobil, in spontaneous speech not all utterances fit adequately into the standard dialogue model because of missing information or unexpected input in addition to misrecognition. Moreover, updating context based on the current dialogue state without an evaluation of the current state cannot reduce cumulative error for future predictions and is likely to introduce cumulative error into the context.

3 System Description

Enthusiast is composed of four main modules: speech recognition, parsing, discourse processing, and generation. Each module is domain-independent and language-independent but makes use of domain specific and language specific knowledge sources for customization.

The hypothesis produced by the speech recognizer about what the speaker has said is passed to the parser. The GLR* parser [4] produces a set of one or more meaning representation structures which are then processed by the discourse processor. The output of the parser is a representation of the meaning of the speaker's sentence. Our meaning representation, called an interlingua text (ILT), is a frame-based language-independent meaning representation. The main component of an ILT are the speech act (e.g., `suggest`, `accept`, `reject`), the sentence type (e.g., `state`, `query-if`, `fragment`), and the main semantic frame (e.g., `free`, `meet`). An example of an ILT is shown in Figure 1.

Development of our discourse processing module was based on a corpus of 20 spontaneous Spanish scheduling dialogues containing a total of 630 utterances.

YO PODRÍA MARTES EN LA MAÑANA
(I could meet on Tuesday in the morning)

```
((SENTENCE-TYPE *STATE)
 (FRAME *MEET)
 (SPEECH-ACT *SUGGEST)
 (A-SPEECH-ACT (*MULTIPLE* *SUGGEST *ACCEPT
                     *STATE-CONSTRAINT))
 (WHO ((FRAME *I)))
 (WHEN
   ((WH -) (FRAME *SIMPLE-TIME)
         (DAY-OF-WEEK TUESDAY)
         (TIME-OF-DAY MORNING)))
 (ATTITUDE *POSSIBLE))
```

Fig. 1. An Interlingua Text (ILT)

We identify a total of fourteen possible speech acts in the appointment scheduling domain [8] (Figure 2). The discourse processing module disambiguates the speech act of each utterance, updates a dynamic memory of the speakers' schedules, and incorporates the utterance into discourse context.

Speech Act	**Example Utterance**
Accept	Thursday I'm free the whole day.
Acknowledge	OK, I see.
Address	Wait, Alex.
Closing	See you then.
Confirm	You are busy Sunday, right?
Confirm-Appointment	So Wednesday at 3:00 then?
Deliberation	Hm, Friday in the morning.
Opening	Hi, Cindy.
Reject	Tuesday I have a class.
Request-Clarification	What did you say about Wednesday?
Request-Response	What do you think?
Request-Suggestion	What looks good for you?
State-Constraint	This week looks pretty busy for me.
Suggest	Are you free on the morning of the eighth?

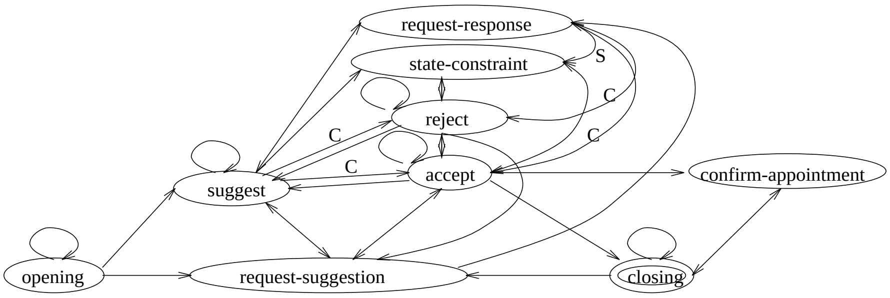
Fig. 2. Speech Acts Covered by Enthusiast

We use four processing components for speech act recognition: a grammar prediction component, a statistical component, a finite state machine, and a plan-based discourse processor. The grammar prediction component assigns a set of possible speech acts to an ILT based on the syntactic and semantic information in the interlingua representation. The resulting set of possible speech acts is inserted into the `a-speech-act` slot of the ILT (See Figure 1). The final determination of the communicative function of the ILT, the speech act, is done by the other three components. The statistical component predicts the following speech act using knowledge about speech act frequencies in our training corpus. The statistical component is able to provide ranked predictions in a fast and efficient way. To cater to the sparse data problem, bigram speech act probabilities are smoothed based on backoff models [12]. The finite state machine (FSM) describes representative sequences of speech acts in the scheduling domain. It is used to record the standard dialogue flow and to check whether the predicted speech act follows idealized dialogue act sequences. The FSM consists of states and transition arcs. The states represent speech acts in the corpus. The transitions between states can have the symbols: `S` (for the same speaker), `C` (for change of speaker), or null (no symbol); the null symbol represents the cases in which the transition is legal, independent of whether the speaker changes or remains the same[4]. A graphical representation of the major parts of the FSM appears in Figure 3. We extended the FSM so that at each state of the finite state machine we allow for phenomena that might appear anywhere in a dialogue, such as `acknowledge`, `address`, `confirm`, `request-clarification`, and `deliberation`. The plan-based discourse processor handles knowledge-intensive tasks exploiting various knowledge sources, including the grammar component predictions and linguistic information. Details about the plan-based discourse processor can be found in [8]. The finite state machine and the statistical component have recently been implemented as a fast and efficient alternative to the more time-consuming plan-based discourse processor. In our future design of the discourse processing module, we may adopt a layered architecture similar to the one proposed in Verbmobil. In such an architecture, the finite state machine would constitute a lower layer providing an efficient way of recognizing speech acts, while the plan-based discourse processor, at a higher layer, would be used to handle more knowledge intensive processes, such as recognizing doubt or clarification sub-dialogues and robust ellipsis resolution. In this paper, we discuss the cumulative error problem in the context of the finite state machine and the statistical component.

4 Speech Act Recognition

For the task of speech act recognition, we use a combination of grammatical, statistical, and contextual knowledge. The finite state machine encodes the preceding context state, tests the consistency of the incoming utterance with the dialogue model and updates the current state. Given the current state, the finite state machine can provide a set of speech acts that are likely to follow. The speech act of the following input utterance should be a member of this set if the input utterance follows the standard dialogue flow. This set of speech acts is compared with the set of possible speech acts (`a-speech-act`) proposed by the grammar component for the same input utterance. The intersection of the finite state machine predictions and the grammar component predictions should yield

[4] Our corpus analysis showed that certain dialogue act sequences are possible only for the same speaker and others are possible only for different speakers.



The state **opening** is the initial state. The state **closing** is the final state.
All other states are non-final states.

Fig. 3. The Main Component of the Finite State Machine

the speech acts which are consistent both with the input semantic representation and with the standard dialogue flow. Oftentimes, an utterance can perform more than one legal function. Bigram speech act probabilities are then used to select the most probable one from the intersection set.

An empty intersection between the two sets of predictions signals an inconsistency between the non-context-based grammar predictions and the context-based FSM predictions. The inconsistency can result from unexpected inputs, missing information, out of domain utterances, or simply misrecognized speech act. We tested two approaches for dealing with the conflicting predictions: a *jumping context approach* and a *hypothesis tree approach*. We describe the two approaches below.

Jumping context approach

The rationale behind the jumping context approach is that while we recognize the predictive power of a statistical model, a finite state machine, or a plan-based discourse processor, we abandon the assumption that dialogue act sequences are always ideal in spontaneous speech. Instead of trying to incorporate the current input into the dialogue context, we accept that speech act sequences can at times be imperfect. Instead of following the expectations provided by the context, we assume there is an inaccurate context and there is a need to re-establish the state in the discourse context. In such cases, we trust the grammar predictions more, assessing the current position using syntactic and semantic information. When there is more than one speech act proposed by the grammar component, we use speech act unigrams to choose the most likely one in the corpus. The context state will then be updated accordingly using the grammar prediction. In the graph representation of the finite state machine, this corresponds to allowing empty arc jumps between any two states. Note that this jump from one state to another in the finite state machine is *forced* and *abnormal* in the sense that it is designed to cater to the abrupt change of the flow of dialogue act sequences in spontaneous speech. Thus it is different from transitions with null symbols, which record legal transitions between states. The algorithm for this approach is described in Figure 4. We demonstrate later that this approach gives better performance than one which trusts context in the case of conflicting predictions.

```
context-state = 'start
FOR each input ilt
  context-predictions = predictions from the FSM
                        given context-state
  grammar-predictions = a-speech-act in input ilt
  Intersect context- and grammar-predictions
  IF intersection is not empty,
    use bigrams to rank the speech acts in intersection
    return the most probable follow up speech act
  ELSE ;;; use grammar predictions
    IF more than one speech act in a-speech-act
      use unigrams to rank the possible speech acts
      return the most probable speech act
    ELSE return a-speech-act
  update context-state using the returned speech act
```

Fig. 4. Algorithm for Jumping Context Approach

As an example, consider the dialogue excerpt in Table 1. After speaker S2 accepts S1's suggestion and tries to close the negotiation, S2 realizes that they have not decided on where to meet. The utterance *no* after the closing *chau* does not fit into the dialogue model, since the legal dialogue acts after a `closing` are `closing`, `confirm-appointment` or `request-suggestion` (see Figure 3). When the standard dialogue model is observed (marked by *Strict Context* in Table 1), the utterance *no* is recognized as `closing` since `closing` is the most probable speech act following the previous speech act `closing`. If upon seeing this conflict we instead trust the grammar prediction (marked by *Jumping Context* in Table 1), by recognizing *no* as a `reject`, we bring the dialogue context back to the stage of negotiation. Trusting the grammar, however, does not imply that we should abandon context altogether. In the test set in the experiments discussed in the next section, context represented by the FSM has shown to be effective in reducing the number of possible speech acts assigned by the grammar component.

Hypothesis tree approach

The rationale behind the hypothesis tree approach is that instead of producing a single speech act hypothesis at the time an utterance is passed to the discourse processor, we delay the decision until a later point. In doing so, we hope to reduce cumulative error due to misrecognition because of early commitment to a decision. Specifically, we keep a set of possible speech act hypotheses for

Dialogue Utterances	Strict Context	Jumping Context
S1: QUÉ TE PARECE EL LUNES NUEVE ENTONCES (HOW IS MONDAY THE NINTH FOR YOU THEN)	suggest	suggest
S2: PERFECTO (PERFECT)	accept	accept
CHAU (BYE)	closing	closing
NO (NO)	closing	reject
ESPÉRATE (WAIT)	address	address
NO (NO)	closing	reject
ESPÉRATE (WAIT)	address	address
ALGO PASÓ MAL (SOMETHING BAD HAPPENED)	no parse	no parse
DÓNDE NOS VAMOS A ENCONTRAR (WHERE ARE WE GOING TO MEET)	request-suggestion	request-suggestion
S1: NO (NO)	reject	reject
ESPÉRATE (WAIT)	address	address
SÍ (YES)	acknowledge	acknowledge
DÓNDE NOS ENCONTRAMOS (WHERE ARE WE MEETING)	request-suggestion	request-suggestion

Table 1. An Example Dialogue

each input utterance as contextual states for future predictions. Each context state may in turn be followed by more than one speech act hypothesis for the subsequent utterance, thus yielding a tree of possible sequences of speech act hypotheses. The hypothesis tree is expanded within a beam so that only a certain total number of branches are kept to avoid memory explosion. When the turn shifts between speakers, the hypothesis path with the highest probability (calculated by multiplying speech act bigram probabilities in that path) is chosen as the best hypothesis for the sequences of ILTs in that turn. Each ILT is then updated with its respective speech act in the chosen hypothesis. For each new turn, the last context state in the best hypothesis of the previous turn is used as the starting root for building new hypothesis tree. Figure 5 gives the algorithm for the hypothesis tree approach.

As in the jumping context approach, the predictions of speech acts for each utterance are the combined result of the context-based FSM predictions and non-context-based grammar predictions. The intersection of both predictions gives the possible speech acts which are consistent with both the dialogue model and the default functions of the input utterance. When there is no intersection, we face the decision of trusting the context-based FSM predictions or the non-context-based grammar predictions. We demonstrate later that, for the hypothesis tree approach, again, trusting grammar predictions gives better results than strictly following context predictions at the time of conflicting predictions.

```
hypothesis-tree = '(((start)))
ILTS = nil
FOR each input ilt
  IF still in the same turn
    push ilt into ILTs
    FOR each path in the hypothesis-tree
      context-state = last state in the path
      get speech act predictions for input ilt
      update hypothesis-tree
  ELSE ;;; turn shifts
    choose the path with the highest probability
    update ilts in ILTS with their respective speech act
      prediction in the chosen path
    ILTS = nil
    context-state = last state in the chosen path
    hypothesis-tree = (((context-state)))
    push ilt into ILTS
    get speech act predictions for input ilt
    update hypothesis-tree
  rank the paths in the hypothesis-tree and
  trim the tree within a beam.
```

Fig. 5. Algorithm for the Hypothesis Tree Approach

5 Evaluation

We developed the finite state machine and the statistical module based on the corpus of 20 dialogues mentioned in Section 3. We tested them on another 10 unseen dialogues, with a total of 506 dialogue utterances. Each utterance in both the training and testing dialogues is tagged with a hand-coded target speech act for the utterance. Out of the 506 utterances in the test set, we considered only the 345 utterances that have possible speech acts (in the `a-speech-act` slot) proposed by the non-context-based grammar component.[5]

We conducted two tests on the set of 345 utterances for which the `a-speech-act` slot is not empty. Test 1 was done on a subset of them, consisting of 211 dialogue utterances for which the grammar component returns multiple possible speech acts: we measured how well the different approaches correctly disambiguate the multiple speech acts in the `a-speech-act` slot with respect to the

[5] For 161 utterances, the grammar component doesn't return any possible speech act. This is because the parser does not return any parse for these utterances or the utterances are fragments. Although it is possible to assign speech acts to the fragments based on contextual information, we found that, without adequate semantic and prosodic information, the context predictions for these fragments are usually not reliable.

hand-coded target speech act. Test 2 was done on the whole set of 345 utterances, measuring the performance of the different approaches on the overall task of recognizing speech acts.

We evaluate the performance of our proposed approaches, namely the jumping context approach and the hypothesis tree approach, in comparison to an approach in which we always try to incorporate the input utterance into the discourse context (marked by *Strict Context* in Table 2). These approaches are all tested in the face of cumulative error[6]. We also measured the performance of randomly selecting a speech act from the `a-speech-act` slot in the ILT as a baseline method. This method gives the performance statistic when we do not use any contextual information provided by the finite state machine.

Approaches	**Test 1**	**Test 2**
Random from Grammar	38.6%	60.6%
Strict Context (Trusting FSM)	52.4%	65.5%
Jumping Context (Trusting Grammar)	55.2%	71.3%
Hypothesis Tree Trusting FSM	48.0%	56.5%
Hypothesis Tree Trusting Grammar	50.0%	60.6%

Table 2. Evaluation: Percent Correct Speech Act Assignments

Table 2 gives some interesting results on the effect of context in spoken discourse processing. Since Test 1 is conducted on utterances with multiple possible speech acts proposed by the non-context-based grammar component, this test evaluates the effects on speech act disambiguation by different context-based approaches. All four approaches employing context perform better than the non-context-based grammar predictions. Test 1 also demonstrates that it is imperative to estimate context carefully. Our experiments show that when context-based predictions and non-context-based predictions are inconsistent with each other, trusting the non-context-based grammar predictions tend to give better results than trusting context-based FSM predictions. In particular, the jumping context approach gives 2.8% improvement over the strict context approach in which context predictions are strictly followed, and trusting grammar predictions gives 2% improvement over trusting FSM predictions in the hypothesis tree approach. To our surprise, the jumping context approach and the strict context approach do better than the hypothesis tree approaches in which more contextual information is available at decision time. This seems to suggest that keeping more contextual information for noisy data, such as spontaneous speech, may actually increase the chances for error propagation, thus making cumulative

[6] We found it hard to test in absence of cumulative error. Because of missing information and unexpected input, it is hard even for the human coder to provide an accurate context.

error a more serious problem. In particular, at the point where grammar and context give conflicting predictions, the target speech act may have such a low bigram probability with respect to the given context state that it gives a big penalty to the path of which it is a part.

Test 2 is conducted on utterances with either ambiguous speech acts or unambiguous speech acts proposed by the grammar component. When an ILT has one unambiguous possible speech act, we can assume that the grammar component is highly confident of the speech act hypothesis, based on the syntactic and semantic information available[7]. Note again that the jumping context approach does better than the strict context approach for dealing with conflicting predictions. The hypothesis tree approach, however, does not improve over the non-context-based grammar approach, regardless of whether the grammar predictions are trusted or the context predictions are trusted. This observation seems to support our belief that reestablishing a context state in case of prediction conflicts is an effective approach to reducing cumulative error. Keeping a hypothesis tree to store more contextual information is not as effective as reestablishing the context state, since more contextual information cannot stop error propagation. As decisions are made at a later point, certain target speech acts may be buried in a low probability path and will not be chosen.

6 Conclusion

In this paper we have discussed our effort to minimize the effect of cumulative error in utilizing discourse context. We challenged the traditional assumption that every utterance in a dialogue adheres to the dialogue model and that the process of recognizing speech acts necessarily results in a speech act that can be best incorporated into the dialogue model. We showed that in spontaneous speech, the ideal dialogue flow is often violated by unexpected input, missing information or out of domain utterances in addition to misrecognition. To model dialogue more accurately, this fact should be taken into account. We experimented with two approaches to reducing cumulative error in recognizing speech acts. Both approaches combine knowledge from dialogue context, statistical information, and grammar prediction. In the case of a prediction conflict between the grammar and the context, instead of blindly trusting the predictions from the dialogue context, we trust the non-context-based grammar prediction. Our results demonstrate that reestablishing a context state by trusting grammar predictions in case of prediction conflicts is more robust in the face of cumulative error. Our future work includes exploring different smoothing techniques for the context model in order to quantify the effectiveness of context in different situations.

[7] The fact that in 134 utterances there is no speech act ambiguity explains the good performance of the random approach.

7 Acknowledgements

We would like to thank Alex Waibel for discussions and comments over this work and thank Chris Manning, Xiang Tong and the reviewers for references and comments. This work was made possible in part by funding from the U.S. Department of Defense. While this work was carried out, the second author was with the Computational Linguistics Program, Carnegie Mellon University.

References

1. J. F. Allen and L. K. Schubert. *The Trains Project.* University of Rochester, School of Computer Science, 1991. Technical Report 382.
2. K. W. Church and W. A. Gale. Probability Scoring for Spelling Correction. *Statistics and Computing*, 1:93–103, 1991.
3. L. Lambert. *Recognizing Complex Discourse Acts: A Tripartite Plan-Based Model of Dialogue.* PhD thesis, Department of Computer Science, University of Delaware, 1993.
4. A. Lavie. *A Grammar Based Robust Parser For Spontaneous Speech.* PhD thesis, School of Computer Science, Carnegie Mellon University, 1995.
5. L. Levin, O. Glickman, Y. Qu, D. Gates, A. Lavie, C. P. Rosé, C Van Ess-Dykema, and A. Waibel. Using Context in Machine Translation of Spoken Language. In *Theoretical and Methodological Issues in Machine Translation*, 1995.
6. Y. Qu, C. P. Rosé, and B. Di Eugenio. Using Discourse Predictions for Ambiguity Resolution. In *Proceedings of the COLING*, 1996.
7. N. Reithinger and E. Maier. Utilizing Statistical Dialogue Act Processing in Verbmobil. In *Proceedings of the ACL*, 1995.
8. C. P. Rosé, B. Di Eugenio, L. S. Levin, and C. Van Ess-Dykema. Discourse Processing of Dialogues with Multiple Threads. In *Proceedings of the ACL*, 1995.
9. R. W. Smith, D. R. Hipp, and A. W. Biermann. An Architecture for Voice Dialogue Systems Based on Prolog-style Theorem Proving. *Computational Linguistics*, 21(3):218–320, 1995.
10. B. Suhm, L. Levin, N. Coccaro, J. Carbonell, K. Horiguchi, R. Isotani, A. Lavie, L. Mayfield, C. P. Rosé, C. Van-Ess Dykema, and A. Waibel. Speech-Language Integration in a Multi-Lingual Speech Translation System. In *Proceedings of the AAAI Workshop on Integration of Natural Language and Speech Processing*, 1994.
11. A. Lavie, D. Gates, M. Gavaldá, L. Mayfield, A. Waibel and L. Levin. Multilingual Translation of Spontaneously Spoken Language in a Limited Domain. In *Proceedings of the COLING*, 1996.
12. S. M. Katz. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. In *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1987.