Dialogue Processing in a Conversational Speech Translation System

Alon Lavie, Lori Levin, Yan Qu, Alex Waibel, Donna Gates
Marsal Gavaldà, Laura Mayfield, and Maite Taboada

Center for Machine Translation
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
email : lavie@cs.cmu.edu

ABSTRACT

Attempts at discourse processing of spontaneously spoken dialogue face several difficulties: multiple hypotheses that result from the parser's attempts to make sense of the output from the speech recognizer, ambiguity that results from segmentation of multi-sentence utterances, and cumulative error — errors in the discourse context which cause further errors when subsequent sentences are processed. In this paper we will describe our robust parsers, our procedures for segmenting long utterances, and two approaches to discourse processing that attempt to deal with ambiguity and cumulative error.

1. Introduction

In this paper we describe how the JANUS [10] multi-lingual speech-to-speech translation system addresses problems that arise in discourse processing of spontaneous speech. The analysis of spoken dialogues requires discourse processors that can deal with ambiguity and cumulative error – errors in the discourse context which cause further errors when subsequent sentences are processed.

The input to our discourse processing module is a set of interlingua texts (ILTs) which are output by the parser. In an attempt to achieve both robustness and translation accuracy when faced with speech disfluencies and recognition errors, we use two different parsing strategies: the GLR* parser designed to be more accurate, and the Phoenix parser designed to be more robust. For both parsers, segmentation into units of coherent meaning is achieved in a two-stage process, partly prior to and partly during parsing. The parsers are described in Section 2 and the segmentation procedures are described in Section 3.

We have also experimented with two approaches to discourse processing: a plan inference approach, designed to keep a detailed representation of the discourse context, and a finite state processor augmented with a statistical component, designed to be fast. Both discourse processors are robust over spontaneous speech. The plan inference system uses graded constraints to assign penalties instead of failing on unexpected input. The finite state approach incorporates a solution to the cumulative error problem. The discourse processors and an evaluation of their performance in assigning speech acts are presented in Section 4. Much of our current research deals with combining the discourse processors with the other translation components in a way that achieves optimal performance. This is described in Section 4.1.

```
((frame *free)
 (who ((frame *i)))
 (when ((frame *simple-time)
        (day-of-week wednesday)
        (time-of-day morning)))
 (a-speech-act (*multiple* *suggest *accept))
 (sentence-type *state)))

Sentence: I could do it Wednesday morning too.
```

Figure 1: An Example ILT

2. The Robust GLR and Phoenix Translation Modules

JANUS employs two robust translation modules with complementary strengths. The GLR module gives more complete and accurate translations whereas the Phoenix module is more robust over the disfluencies of spoken language. The two modules can run separately or can be combined to gain the strengths of both.

The GLR module is composed of the GLR* parser [2][3], the LA-Morph morphological analyzer and the GenKit generator. The GLR* parser is based on Tomita's Generalized LR parsing algorithm [8]. GLR* skips parts of the utterance that it cannot incorporate into a well-formed sentence structure. Thus, it is well-suited to domains in which non-grammaticality is common. The parser conducts a search for the maximal subset of the original input that is covered by the grammar. JANUS GLR grammars are designed to produce feature structures that correspond to a frame-based language-independent representation of the meaning of the input utterance. For a given input utterance, the parser produces a set of interlingua texts, or ILTs. An example of an ILT is shown in Figure 1. The GLR* parser also includes several tools designed to address the difficulties of parsing spontaneous speech, including a statistical disambiguation module, a self-judging parse quality heuristic, and the ability to segment multi-sentence utterances.

The JANUS Phoenix translation module [4] is an extension of the Phoenix Spoken Language System [9]. It consists of a parsing mod-

Original utterance:
SÍ QUÉ TE PARECE TENGO EL MARTES DIECIOCHO Y EL MIÉRCOLES DIECINUEVE LIBRES TODO EL DÍA PODRÍAMOS IR DE MATINÉ O SEA EN LA TARDE VER EL LA PELÍCULA
(Roughly "*Yes what do you think I have Tuesday the eighteenth and Wednesday the nineteenth free all day we could go see the matinée so in the afternoon see the the movie.*")

As decoded by the recognizer:
%NOISE% SI1 QUE1 TE PARECE %NOISE% TENGO EL MARTES DIECIOCHO Y EL MIE1RCOLES DIECINUEVE LIBRES TODO EL DI1A PODRI1AMOS IR DE MATINE1 %NOISE% O SEA LA TARDE A VER LA

Parsed:
%<S> si1 que1 te parece tengo el martes dieciocho y el mie1rcoles diecinueve libres todo el di1a podri1amos *IR *DE -MATINE1 o sea la tarde a ver LA %</S>

Parse Tree (≡ Semantic Representation):

```
[respond] ( [yes] ( SI1 ))

[your_turn] ( QUE1 TE PARECE )

[give_info] ( [my_availability] ( TENGO [temp_loc]
( [temporal] ( [point] ( [date] ( EL [d_o_w] ( MARTES ))
[date] ( [day_ord] ( DIECIOCHO ) [conj] ( Y ) EL [d_o_w]
( MIE1RCOLES )) [date] ( [day_ord] ( DIECINUEVE )))))
LIBRES ))

[give_info] ( [my_availability] ( [temp_loc]
( [temporal] ( [range] ( [entire] ( TODO )EL [unit]
( [t_unit] ( DI1A )))))PODRI1AMOS ))

[suggest] ( [suggest_meeting] ( [temp_loc] ( [temporal]
( O SEA [point] ( LA [t_o_d] ( TARDE ))))A VER ))
```

Generated:
English = <Yes what do you think? I could meet Tuesday eighteenth and Wednesday the nineteenth I could meet the whole day do you want to try to get together in the afternoon>

Figure 2: A Phoenix Spanish-to-English Translation Example

ule and a generation module. Unlike the GLR method which attempts to construct a detailed ILT for a given input utterance, the Phoenix approach attempts to only identify the key semantic concepts represented in the utterance and their underlying structure. It allows the ungrammaticalities that often occur between phrases to be ignored and reflects the fact that syntactically incorrect spontaneous speech is often semantically well-formed. An example of output from the Phoenix parser is shown in Figure 2. The parsed speech recognizer output is shown with unknown (-) and unexpected (*) words marked. These segments were ignored by the parser.

The Phoenix parsing grammar specifies patterns which represent concepts in the domain. Each concept, irrespective of its level in the hierarchy, is represented by a separate grammar file. These grammars are compiled into Recursive Transition Networks (RTNs). The parser matches as much of the input utterance as it can to the patterns specified by the RTNs. The parser can ignore any number of words in between top-level concepts, handling out-of-domain or otherwise unexpected input. The parser has no restrictions on the order in which slots can occur. This may add to the ambiguity in the segmentation of the utterance into concepts. The parser uses a disambiguation algorithm that attempts to cover the largest number of words using the smallest number of concepts. The result is a meaningful but somewhat telegraphic translation.

Although both GLR* and Phoenix were specifically designed to deal with spontaneous speech, each of the approaches has some clear strengths and weaknesses. Because each of the two translation methods appears to perform better on different types of utterances, they may hopefully be combined in a way that takes advantage of the strengths of each of them. One strategy that we have investigated is to use the Phoenix module as a back-up to the GLR module. The parse result of GLR* is translated whenever it is judged by a parse quality heuristic to be "Good". Whenever the parse result from GLR* is judged as "Bad", the translation is generated from the corresponding output of the Phoenix parser. Results of using this combination scheme are presented in Section 4.2. We are in the process of investigating some more sophisticated methods for combining the two translation approaches.

3. Segmentation

Spoken utterances are often composed of several sentences and/or fragments. Our interlingual approach to translation requires that utterances be broken down into units of coherent meaning or discourse function. We call these units Semantic Dialogue Units (SDUs). Utterance segmentation in our system is a two stage process. In the first stage, the utterance is broken down into smaller segments or "chunks" based on acoustic, statistical and lexical cues. The smaller segments are then passed on to the parsers, which further segment them into SDUs using their own internal criteria.

The acoustic cues we use in the pre-parsing segmentation procedure include silence information and human and non-human noises which we have found to be indicative of some SDU boundaries. The statistical component of the segmentation procedure is a confidence measure that attempts to capture the likelihood of a SDU boundary between any pair of words in the utterance. Assume these words are $[w_1 w_2 \bullet w_3 w_4]$, where the potential SDU boundary being considered is between w_2 and w_3. The likelihood of an SDU boundary at this point is determined using an estimated probability that is based on a combination of three bigram frequencies: $F([w_1 w_2 \bullet])$, $F([w_2 \bullet w_3])$ and $F([\bullet w_3 w_4])$, representing the frequency of an SDU boundary occuring to the right, in between, or to the left of the appropriate bigram. Breaks are predicted at points where the estimated probability exceeds a threshold that was arrived at experimentally. The third component of the pre-parsing segmentation procedure is a set of lexical cues. These cues are language- and domain-specific words or phrases that have been determined through linguistic analysis to have a very high likelihood of preceding or following an SDU boundary. These phrases alone do not trigger SDU boundary breaks. They are combined with the statistical component. The occurrence of a lexical cue triggers a "boost" increment to the probability of an SDU boundary, as determined by the statistical component.

4. Discourse Processing

The discourse processing module in Janus disambiguates the speech act of each SDU, updates a dynamic memory of schedules, and incorporates the SDU into discourse context. We have experimented with two approaches to discourse processing: a plan inference system (based on work by Lambert [1]) and a finite state processor

```
Unsegmented Speech Recognition:

  (%noise% si1 mira toda la man5ana estoy disponible
   %noise% %noise% y tambie1n el fin de semana si podri1a
   hacer mejor un di1a fin de semana porque justo el once
   no puedo me es imposible va a poder fin de semana)

Pre-broken Speech Recognition:

  (si1)
  (mira toda la man5ana estoy disponible %noise% %noise%
   y tambie1n el fin de semana)
  (si podri1a hacer mejor un di1a fin de semana)
  (porque justo el once no puedo me es imposible va a
   poder fin de semana)

Parser SDU Segmentation (of Pre-broken Input):

 (((si1))
  ((mira) (toda la man5ana estoy disponible) (y tambie1n)
   (el fin de semana))
  ((si podri1a hacer mejor un di1a fin de semana))
  ((porque el once no puedo) (me es imposible)
   (va a poder fin de semana)))

Translation:

  "yes --- Look all morning is good for me -- and also
   -- the weekend --- If a day weekend is better ---
   because on the eleventh I can't meet --
   That is bad for me can meet on weekend"
```

Figure 3: Segmentation of a Spanish Full Utterance

augmented with a statistical component. The plan-based approach handles knowledge-intensive tasks, exploiting various knowledge sources. The finite state approach provides a fast and efficient alternative to the more time-consuming plan-based approach. Currently, the two discourse processors are used separately. We intend to combine these two approaches with a layered architecture, similar to the one proposed for Verbmobil [6], in which 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. The performance of each approach in assigning speech acts is presented in Section 4.2.

The plan-based discourse processor [7] takes as its input the best parse returned by the parser. The discourse context is represented as a plan tree. The main task of the discourse processor is to relate the input to the context, or the plan tree. In general, plan inference starts from the surface forms of sentences from which speech acts are then inferred. Multiple speech acts can be inferred for one ILT. A separate inference chain is created for each potential speech act performed by the associated ILT. Preferences for picking one inference chain over another were determined by a set of focusing heuristics, which provide ordered expectations of discourse actions given the existing plan tree. The speech act is recognized in the course of determining how the inference chain attaches to the plan tree.

The finite state machine (FSM) discourse processor [5] 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 states in the FSM represent speech acts in the domain. The transitions between states record turn-taking information. Given the current state, multiple following speech acts are possible. The statistical component (consisting of speech act n-grams) is used to provide ranked predictions for the following speech acts.

One novel feature of the finite state approach is that we incorporate a solution to the cumulative error problem. 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. 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. To reduce cumulative error, we focus on instances of conflict between the predictions of the FSM and the grammar. Our experiments show that in the case of a prediction conflict between the grammar and the FSM, instead of blindly trusting the predictions from the dialogue context, trusting the non-context-based grammar predictions gives better performance in assigning speech acts. This corresponds to a *jump* from one state to another in the finite state machine. Section 4.2 reports the performance of the FSM with jumps determined by the non-context-based predictions of the grammar.

4.1. Late Stage Disambiguation

The robust parsing components discussed in Section 2 employ a large flexible grammar to handle such features of spoken language as speech disfluencies, speech recognition errors, and the lack of clearly marked sentence boundaries. This is necessary to ensure the robustness and flexibility of the parser. However, as a side-effect, the number of ambiguities increases. An important feature of our approach to reducing parse ambiguity is to allow multiple hypotheses to be processed through the system, and to use context to disambiguate between alternatives in the final stages of the processing, where knowledge can be exploited to the fullest. Local utterance-level predictions are generated by the parser. The larger discourse context is processed and maintained by the discourse processing component, which has been extended to produce context-based predictions for resolving ambiguity. The predictions from the context-based discourse processing approach and those from the non-context-based parser approach are combined in the final stage of processing.

We experimented with two methods of automatically learning functions for combining the context-based and non-context-based scores for disambiguation, namely a genetic programming approach and a neural net approach. While we were able, in the absence of cumulative error, to get an improvement of both combination techniques over the parser's non-context-based statistical disambiguation technique, in the face of cumulative error, the performance decreased significantly. We are in the process of incorporating our cumulative error reduction technique in the task of disambiguation.

Approaches	Per cent correct
Random from Grammar	38.6%
FSM Strict Context	52.4%
FSM Jumping Context	55.2%
Plan-Based DP	53.8%

Table 1: Approaches to Speech Act Assignment

4.2. Evaluation

The results in Table 1 show the the performance of the two discourse processing approaches, namely the plan-based approach and the finite state machine approach for the task of assigning speech acts. The FSM processor with the cumulative error reduction mechanism is marked by *FSM Jumping Context*, and the FSM without jumping is marked by *FSM Strict Context*. The choice of randomly selecting a speech act from the non-context-based predictions of the grammar indicates the performance of the system when we do not use any contextual information.

We tested the discourse processors on ten unseen dialogues, with a total of 506 utterances. Out of the 506 utterances in the test set, we considered only 211 utterances for which the grammar returns multiple possible speech acts. We measured how well the different approaches correctly disambiguate the multiple speech acts with respect to hand-coded target speech acts.

Table 1 demonstrates the effect of context in spoken discourse processing. Since the test was conducted on utterances with multiple possible speech acts proposed by the non-context-based grammar component, it evaluates the effectiveness of the various context-based approaches in disambiguating speech acts. All of the approaches employing context perform better than the non-context-based grammar predictions. The evaluation also demonstrates that it is imperative to estimate context carefully. The FSM jumping context approach, which attempts to reduce cumulative error, gives better performance than the the FSM strict context approach. It is even better than the more knowledge-intensive plan-based approach. We expect that performance of plan-based approach will improve when we introduce a solution to the cumulative error problem.

5. Conclusions and Future Work

In this paper, we described how our system addresses problems that arise in discourse processing of spontaneous speech. First, we described two different robust parsing strategies — the GLR* parser and the Phoenix parser, and the procedures that both parsers use to segment the input into units of coherent meaning representation that are also of an appropriate size for discourse processing. Then we described two approaches to discourse processing — the plan inference approach and the finite state approach. In describing the finite state approach, we presented one solution to the cumulative error problem. Finally, we described our method of late stage disambiguation where the context-based predictions from our discourse processors are combined with the non-context-based predictions from the parsers. Our future efforts will concentrate on finding improved methods for combining different knowledge sources effectively for the disambiguation task, treating cumulative error in the plan-based discourse processor, and improving the effectiveness of contextual information in constraining the speech translation process.

Acknowledgements

The work reported in this paper was funded in part by grants from ATR - Interpreting Telecommunications Research Laboratories of Japan, the US Department of Defense, and the Verbmobil Project of the Federal Republic of Germany.

6. REFERENCES

1. L. Lambert. *Recognizing Complex Discourse Acts: A Tripartite Plan-Based Model of Dialogue*. PhD thesis, Department of Computer Science, University of Delaware, 1993.
2. A. Lavie and M. Tomita. *GLR* - An Efficient Noise Skipping Parsing Algorithm for Context Free Grammars,* Proceedings of the third International Workshop on Parsing Technologies (IWPT-93), Tilburg, The Netherlands, August 1993.
3. A. Lavie. An Integrated Heuristic Scheme for Partial Parse Evaluation, Proceedings of the 32nd Annual Meeting of the ACL (ACL-94), Las Cruces, New Mexico, June 1994.
4. L. Mayfield, M. Gavaldà, Y-H. Seo, B. Suhm, W. Ward, A. Waibel. Parsing Real Input in JANUS: a Concept-Based Approach. In *Proceedings of TMI 95*.
5. Y. Qu, B. Di Eugenio, A. Lavie, L. Levin and C. P. Rosé. *Minimizing Cumulative Error in Discourse Context,* To appear in Proceedings of ECAI Workshop on Dialogue Processing in Spoken Language Systems, Budapest, Hungary, August 1996.
6. N. Reithinger and E. Maier. Utilizing statistical dialogue act processing in Verbmobil. In *Proceedings of the ACL*, 1995.
7. C. P. Rosé, B. Di Eugenio, L. Levin and C. Van Ess-Dykema. *Discourse Processing of dialogues with multiple threads,* In Proceedings of the ACL, 1995
8. M. Tomita. An Efficient Augmented Context-free Parsing Algorithm. *Computational Linguistics*, 13(1-2):31–46, 1987.
9. W. Ward. Extracting Information in Spontaneous Speech. In *Proceedings of International Conference on Spoken Language*, 1994.
10. M. Woszczyna, N. Aoki-Waibel, F. D. Buo, N. Coccaro, T. Horiguchi, K. and Kemp, A. Lavie, A. McNair, T. Polzin, I. Rogina, C. P. Rosé, T. Schultz, B. Suhm, M. Tomita, and A. Waibel. JANUS-93: Towards Spontaneous Speech Translation. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'94)*, 1994.