SPEECH–LANGUAGE INTEGRATION IN A MULTI–LINGUAL SPEECH TRANSLATION SYSTEM

B. Suhm[1], *L. Levin*[1], *N. Coccaro*[1], *J. Carbonell* [1], *K.Horiguchi*[1], *R. Isotani*[2], *A. Lavie*[1], *L. Mayfield* [1], *C. P. Rosé* [1], *C. Van Ess-Dykema*[3] *A. Waibel* [1,4]

[1] Carnegie Mellon University (USA)
[2] ATR Interpreting Telecommunications Laboratory (Japan)
[3] U.S. Department of Defense
[4] Karlsruhe University (Germany)

ABSTRACT

In this paper we report on our efforts to combine speech and language processing toward multi-lingual spontaneous speech translation. The ongoing work extends our JANUS system effort toward handling spontaneous spoken discourse and multiple languages. A major objective of this project is to maximize the number of modules, methods and data structures that are language-independent and extensible to other domains. After an overview of the task, databases and the system architecture we will focus on how speech decoding and natural language processing modules will be integrated in a large-scale multi-lingual speech-to-speech translation system for spontaneous spoken discourse.

1. INTRODUCTION

The goal of the JANUS project is *multi-lingual* machine translation of *spontaneously spoken dialogs* in a limited domain: two people scheduling a meeting with each other. We are currently working with German, Spanish, and English as source languages and German, English, and Japanese as target languages. This paper reports on our efforts to make NLP robust over spontaneous speech and to use NLP to constrain speech recognition. Towards this end we are investigating statistical, connectionist, and knowledge-based approaches to robust parsing and dialog modeling. We must also adapt plan based discourse processing [1, 2] to accommodate less structured negotiation dialogs and integrate it with machine translation [3].

A major objective of this project is to maximize the number of modules, methods and data structures that are language–independent and extensible to other domains. Language-independent modules include the acoustic processing module, the search engine, the robust parser, and the discourse plan tracking module. These are processors that use independently specified knowledge about different languages in order to process those languages. Language-independent methods include data collection protocols, transcription conventions, methods for building of acoustic models and language models, and grammar creation methods. Common data structures include signal representation, language model specification, N-best word lattices, grammar rules, ILT (Interlingua), and discourse plan operators. A complete description of JANUS can be found in [7].

2. THE SCHEDULING TASK DATABASE

To be able to develop a system for spontaneous speech, we have started to collect a large database of human-to-human dialogs on the scheduling task. Several sites in Europe, the US and Japan have now adopted scheduling as a common task under several research projects. These projects include the German Government's Verbmobil project for German and English translation, the Enthusiast project for Spanish-to-English translation supported by the U.S. Department of Defense, and the activities of the C-STAR consortium of companies and universities in the U.S., Germany, and Japan for translation of German, English, and Japanese.

The data collection procedure involves two subjects who are each given a calendar and are asked to schedule a meeting. There are 13 different calendar scenarios differing from each other in what is scheduled and how much overlap there is in the free time of the two participants. Data has been collected in English, German, and Spanish using the same data collection protocols at Carnegie Mellon University, Karlsruhe University, and the University of Pittsburgh.

The advantages of this experimental design using the same calendars for all languages is that it solicits similar domain-limited dialogs while ensuring a spontaneous, natural (not read or contrived) speaking style. Thus techniques can be compared across languages, and have enabled us to explore automatic knowledge-acquisition and MT techniques in several languages on a comparable task. Table 1 specifies the amount of data collected in each language in terms of the number of dialogs and the number of utterances that have been recorded and transcribed.

We have developed standard transcription conventions that are employed across languages, ensuring uniformity and consistency. Words are transcribed into their conventional spelling. The transcription also indicates human non-speech noises, non-human noises, silences, false starts, mispronunciations, and some intonation. A sample of part of a dialogue is given in Figure 1.

	English		German		Spanish	
	dialogs	utterances	dialogs	utterances	dialogs	utterances
recorded	383	4000	451	4628	146	2920
transcribed	328	3300	215	2293	68	1080

Table 1: State of Data Collection March 1994

Speaker 1: /h#/ /um/ when can we get together
again {comma} < on our
[m(eeting)] > {comma} /um/ to discuss
our project {period} {seos}
/um/ how's @how is@ {comma} /um/ Monday
the eighth {quest} around two
thirty {quest}
#key_click# #paper_ruffle# {seos}

Speaker 2: #key_click# /ls/ /h#/ /uh/
Monday afternoon's @afternoon is@
no good {period} {seos}
I've @I have@ got a meeting from two to
four {comma} {seos}
that's @that is@ not gonna @going to@
give us enough time to get together
{comma} {seos}
/h#/ /um/ *pause* Tuesday afternoon {comma}
the ninth {comma}
would be okay for me though {comma}
#key_click# /h#/ {seos}

Speaker 1: /ls/ /h#/ unfortunately I'll @I will@
be out of town {comma}
from {comma} the ninth {comma} through
the eleventh {period} {seos}
/um/ checking my calendar {comma}
/im/ /h#/ Friday's @Friday is@ no
good {comma} either {period} {seos}
let's @let us@ see {comma} maybe next week
{comma} {seos} /h#/ /oh/ /h#/ that's
@that is@ bad {comma} {seos}
< my class schedule's @schedule is@
[t] {comma} {seos} >
okay {comma} /h#/ how 'bout on
Tuesday the sixteenth {comma}
any time after twelve thirty period
#key_click# /h#/ /h#/ {seos}

Figure 1: Sample Transcription: Text contained in slashes represent human noise; hash marks–non-human noise; curly braces–intonation (except {seos}); angle brackets–false starts; square brackets–mispronunciations; @–contractions; {seos}–end of semantic sentence unit.

Figure 2 shows the growth of vocabulary as the size of the database increases. The vocabulary for Spanish and German is higher than that of English, presumably because of the greater amount of inflectional morphology.

Recent studies [9] and our own observations show that there is a significantly higher rate of disfluencies in human-human dialogs, compared to human-machine database queries. Table 2 compares disfluencies in human-human spontaneous scheduling tasks (SST) in German, English, and Spanish and human-machine queries (ATIS). The table shows the utterance length in words as well as human noises (filled pauses, laughter, coughs, etc. but not intelligible words such as "okay", "well") and false starts (chopped words and

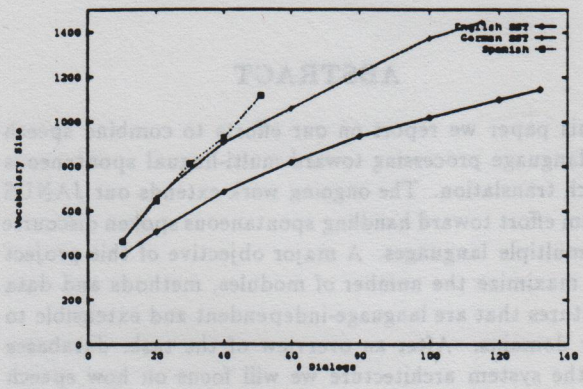

Figure 2: Development of Vocabulary Size

repetitions, deletions, substitutions and insertions of words, but not filled pauses) as percentages of the total number of words in the transcripts [1]. Table 2 suggests that human–human dialogs lead to longer utterances which are more disfluent.

	ATIS	GSST	ESST	SSST
Utterance Length	10	22	30	48
Human Noises	0.5	9.1	13.8	15.7
False Starts	0.9	1.4	1.8	3.5

Table 2: Disfluencies in ATIS vs. Scheduling

In addition, Table 3 in Section 4.2 shows perplexities for bigram and trigram language models for English, Spanish, and German scheduling dialogs as well as English ATIS dialogs. Comparison across languages reveals that spontaneous human-human dialogs yield different perplexities, again presumably due to differing amounts of morphological variation in each language. Comparison with ATIS suggests, that a human-human dialog task (albeit limited) leads to larger perplexities than human-machine database queries.

[1] To exclude artifacts from differing data collection set-ups we didn't consider non-human noises (e.g. clicks, paper rustle) in this statistics.

3. SYSTEM ARCHITECTURE

The main system modules are speech recognition, parsing, discourse processing, and generation.[2] Each module is designed to be language-independent in the sense that it consists of a general processor that applies independently specified knowledge about different languages. Therefore, each module actually consists of a processor and a set of language-specific knowledge sources. A system diagram is shown in Figure 3.

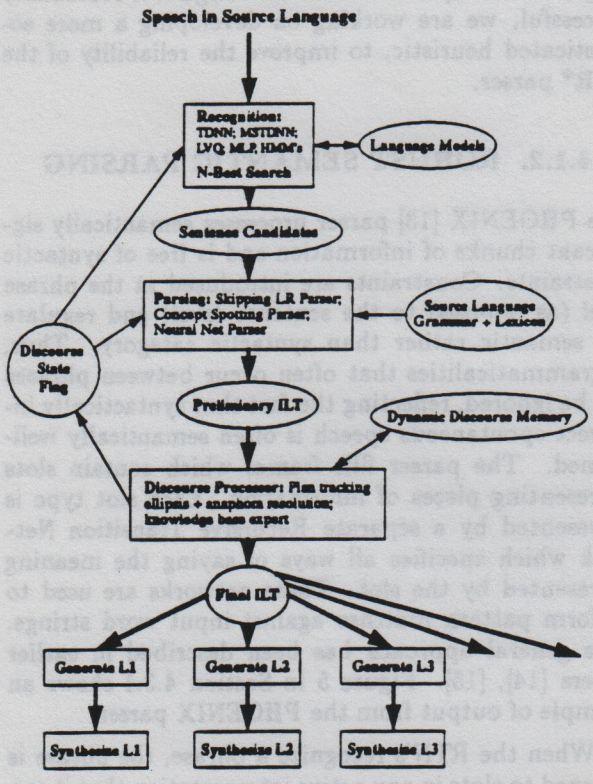

Figure 3: System Diagram

We employ a multi-strategy approach for several of the main processes. For example, we are experimenting with TDNN, MS-TDNN, MLP, LVQ, and HMM's for acoustic modeling; n-grams, word clustering, and automatic phrase detection for language modeling; statistically trained skipping LR parsing, neural net parsing, and robust semantic parsing for syntactic and semantic analysis; and statistical models as well as plan inferencing for identification of the discourse state. The multi-strategy approach should lead to improved performance with appropriate weighting of the output from each strategy.

Processing starts with speech input in the source language. Recognition of the speech signal is done with the acoustic modeling methods mentioned above, constrained by the language model, which is influenced by the current discourse state. This produces a list of the N-best sentence candidates, which are then sent to the translation components of the system.

[2]Discourse processing has not yet been implemented. In this paper we are reporting our plans for this component.

At the core of our machine translation system is an interlingua, which is intended to be a language-independent representation of meaning. The parser outputs a preliminary interlingua text (ILT) or some ILT fragments corresponding the source language input. Multi-sentence conversational turns are assumed to be broken down into separate sentences or sentence fragments before parsing. After parsing, the ILT is further specified by the discourse processor. The discourse processor performs functions such as disambiguating the speech act or discourse function, resolving ellipsis and anaphora, and assembling ILT fragments into full ILTs. It also updates a calendar in the dynamic discourse memory to keep track of what the speakers have said about their schedules. Based on the current discourse state, a flag is set, which is used by the parser to resolve ambiguities in the next sentence to be parsed, and by the recognizer to dynamically adapt the language model to recognize the next utterance. Once the ILT is fully specified, it can be sent to the generator to be rendered in any of the target languages.

4. INTEGRATION OF SPEECH RECOGNITION AND NATURAL LANGUAGE PROCESSING

In building a machine translation system for spontaneously spoken discourse, our tasks include the development of NLP techniques that are robust over speech errors and recognition errors, and the adaptation of speech recognition components to make use of the discourse context and domain-specific knowledge. Because we are working on a multi-lingual task, the techniques we are developing must be generalizable across languages.

4.1. ROBUST PARSING OF SPOKEN INPUT

Coping with spontaneous discourse phenomena and acoustical recognition errors requires robust language parsing. Our approach is to develop language-independent mechanisms, coupled with language and task specific data structures (semantic grammars, augmented by statistical training). Methods employed include extending the Augmented-LR parsing method to work on sentence fragments and to skip over incomprehensible segments, and a pattern-based semantic parser.

4.1.1. THE GLR* PARSER

GLR* is an extended robust version of the Generalized LR Parser, that allows the skipping of unrecognizable parts of the input sentence [10, 11]. It is designed to enhance the parsability of domains such as spontaneous speech, where the input is likely to contain deviations from the grammar due to noise, extra-grammaticalities and limited grammar coverage. If the complete input sentence is not covered by the grammar the parser attempts to find maximal subsets of the input that are

parsable. Some sentences and the corresponding parser output can be seen in Figure 4.

Transcription: Checking my calendar Friday's no good either

```
((ADVERB ALSO)
(WHEN ((FRAME *SIMPLE-TIME) (DAY-OF-WEEK FRIDAY)))
(SPEECH-ACT (*MULTIPLE* *REJECT
    *STATE-CONSTRAINT))
(WHO ((FRAME *I)))
(FRAME *BUSY)
(SENTENCE-TYPE *STATE))
```

Skipped: CHECKING, CALENDAR

Figure 4: Example for the GLR* Skipping Parser

To select the "best" parse we use an integrated evaluation heuristic that combines several different measures, including a statistical component. The parse evaluation heuristic uses a set of features of both the candidate parse and the ignored parts of the original input sentence, by which each of the parse candidates can be evaluated and compared. The features are designed to be general and, for the most part, grammar and domain independent. For each parse, the heuristic computes a penalty score for each of the features. The penalties of the different features are then combined into a single score using a linear combination. The weights used in this scheme are adjustable, and can be optimized for a particular domain and/or grammar.

The current set of evaluation features includes the number and position of skipped words, the fragmentation of the parse analysis and the statistical score of the disambiguated parse tree. Parses that are more fragmented, or require the skipping of more input words receive higher penalties.

The statistical module attached to the parser is similar in framework to the one proposed by Briscoe and Carroll [12], in which shift and reduce actions of the LR parsing tables are directly augmented with probabilities. Training of the probabilities is performed on a set of disambiguated parses. The probabilities of the parse actions induce statistical scores on alternative parse trees, which are then used for parse disambiguation.

However, we also use the statistical component of the parser to evaluate competing parse candidates that correspond to *different* skipped words. The statistical score (sscore) is first converted into a confidence measure, such that more "common" parse trees receive a lower penalty score:

$$penalty = (0.1 * (-log_{10}(sscore)))$$

Thus, a parse candidate with a significantly higher statistical score may be selected, even if it is not maximal in word coverage.

The utility of a parser such as GLR* obviously depends on the semantic coherency of the parse results that it returns. Since the parser is designed to succeed in parsing almost any input, parsing success by itself can no longer provide a likely guarantee of such coherency. We therefore use a filtering heuristic which attempts to filter out incoherent parses. The filtering heuristic attempts to classify the parse chosen as best by the parser into one of two categories: "good" or "bad". The current heuristic is extremely simple and takes into account both the actual value of the parse's combined penalty score and a measure relative to the length of the input sentence. Although it is reasonably successful, we are working on developing a more sophisticated heuristic, to improve the reliability of the GLR* parser.

4.1.2. ROBUST SEMANTIC PARSING

The PHOENIX [13] parser processes semantically significant chunks of information and is free of syntactic constraints. Constraints are introduced at the phrase level (as opposed to the sentence level) and regulate the semantic rather than syntactic category. Thus, ungrammaticalities that often occur between phrases can be ignored, reflecting the fact that syntactically incorrect spontaneous speech is often semantically well-formed. The parser fills frames which contain slots representing pieces of information. Each slot type is represented by a separate Recursive Transition Network which specifies all ways of saying the meaning represented by the slot. These networks are used to perform pattern matches against input word strings. This general approach has been described in earlier papers [14], [15]. Figure 5 in Section 4.3.1 shows an example of output from the PHOENIX parser.

When the RTN's recognize a phrase, the phrase is assigned to slots in any active interpretation that it can fill. If the parser cannot match words between recognized phrases, the words are simply skipped. From the resulting beam of possible interpretations, the highest scoring interpretation is selected when the utterance terminates. The scoring algorithm gives higher scores to interpretations that include more of the input words and to interpretations that use fewer RTNs. The resulting parse is a list of top-level slots, representing for instance statements of availability and suggestions of time, and their associated subnet values (i.e. the actual time that is being suggested). Since information chunks (slots) can stand alone as well as be nested, short sentence fragments are handled in the same way as are semantically correct sentences.

Generation can be accomplished by mapping the resulting parse onto either an ILT, which is then input to a generation grammar, or directly onto patterns in the target language. We have begun work on the latter; each "concept" (slot) has a single phrase translation that is retrieved from a lookup table. Only variables such as numbers are translated directly.

This type of parser is particularly well suited to spontaneous speech, as it ignores most conjunctions

and prepositions which are difficult for the speech recognizer to extract. It is robust over the fragmented sentence structure that native speakers frequently use and produces a meaningful parse by processing only the part of the utterance relevant to the scheduling task.

4.2. LANGUAGE MODELING

In this section we describe methods that attempt to take advantage of natural equivalence word classes and frequently occurring word sequences/phrases, and that also try to take into consideration the acoustic confusability of hypothesized words. The perplexity of these methods was compared on three languages, English, German, and Spanish, using the Spontaneous Scheduling Task databases (ESST, GSST, SSST). As a control experiment, we repeated our experimentation on the well-known ATIS database. In addition, we report preliminary recognition results on the English databases (ESST).

The standard word bigram and, even more, word trigram models need very large databases to obtain robust probability estimates. To collect such large amounts of data is a costly and time consuming process. Models based on word classes can be trained on smaller amounts of data. An automatic word class cluster algorithm was developed to find natural classes of words (see also [16]). Beginning with some initial assignment of words to classes, the clustering algorithm moves words to classes to minimize the perplexity on some development test data. To prevent the optimization from getting stuck in a local optimum, a simulated annealing method is employed.

Beyond word classes, sequences of words can also be bundled into frequently occurring phrases if these reduce test set perplexity. A word phrase bigram language model is proposed to reduce perplexity as well as automatically define common word phrases or idioms in a given task. The resulting word phrases include such common expressions for scheduling dialogs as "out-of-town", "do-you-have", "in-the-next", "what-about", "a-meeting-from".

	ESST	GSST	SSST	ATIS Nov92
Word Bigrams	38	87	72	20
Word Trigrams	35	81	70	15
Cluster Bigrams	37	74	58	20
Word Phrases	35	82	70	18
Cluster Word & Bigrams	34	66	50	19
Cluster Word & Phrases	34	66	49	18
Cluster Word & Trigrams	31	63	46	14

Table 3: Perplexities for ESST, GSST, SSST and ATIS

Table 3 summarizes our results. Compared to a baseline word bigram model, all methods yield perplexity reduction, especially when interpolated with the word bigram model. Some additional improvements can be obtained when using the word trigram model, at the expense of greater computational requirements in the speech decoder. The word phrase model obtains modest perplexity reductions. It can, however, be improved by adding a training method for acoustic models which accounts for coarticulation within word phrases. It can also be used for automatic grammar acquisition. The cluster bigram model obtained the highest perplexity reductions, especially when interpolated with either a word bigram or word phrase model. Although basing models on classes is in general an information-losing process the clustering reduces perplexity because of the sparsity of training data; it can be viewed as a kind of smoothing. As an interesting side effect, words with similar meanings are put in the same clusters.

	Word Bigrams	Cluster Bigrams	Word Phrases
ESST	61	59.2	66.4

Table 4: Preliminary Recognition Results for ESST

Preliminary recognition results were obtained on the English database (ESST). Table 4 shows the word accuracy on a 14 dialog evaluation set. The higher performance of the word phrase model can be attributed to the fact that it builds word phrases including easily misrecognized function words.

4.3. USING DISCOURSE TO CONSTRAIN SPEECH RECOGNITION

In this section we describe the use of discourse knowledge in the system's speech decoder to improve performance. We have conducted two preliminary experiments on statistical dialog modeling, once involving the ouput of the PHOENIX parser and one involving the ILTs that are produced by the GLR* parser. We also describe how linguistic knowledge from a plan inference system can be used to predict possible next utterances.

4.3.1. SLOT-BASED LANGUAGE MODELING

Our first experiment on statistical dialog modeling extends work by Pieraccini et al [17] and Ward [6]. We used the PHOENIX parser to automatically label different parts of a sentence with frames and slot fillers and then trained stochastic models similar to [17]. Instead of using these models to find the conceptual structure of utterances we suggest to dynamically adapt the language model, similar to an approach described in [6].

In the training phase, we extract the sequence of top-level slots and the corresponding sequences of words for each slot from the output of the semantic parser (see Figure 5). A junk slot absorbs all words which are not covered by the semantic grammar.

Transcription: Again Tuesday morning's not very good for me I'm busy from nine to twelve Let's see What about Wednesday on the sixth

give_info	TUESDAY MORNING'S NOT VERY GOOD FOR ME I'M BUSY FROM NINE TO TWELVE
interject	LET'S SEE
suggest_time	WHAT ABOUT WEDNESDAY ON THE SIXTH

Figure 5: Extracting the Slot Sequence from the Semantic Parser's Output

Such sequences of words labeled with their respective slots can be used to estimate slot transition probabilities $P(S_i \mid S_{i-1})$ and slot–dependent word bigram probabilities $P(w_i \mid w_{i-1}, S_i)$.

In the speech decoding process, the language model can be dynamically adapted by interpolating the slot–dependent bigram models according to the current prediction of the next slot $P(S_i \mid S_{i-1})$. One can imagine the search for word sequences as a hidden Markov process with its states being the top–level slots, representing the current dialog state.

In a preliminary experiment, we trained such a model on 143 dialogs of the ESST database and tested it on a 14 dialog evaluation set. Perplexities didn't differ much, but we expect better results by using slot trigrams instead of slot bigrams, changing the definition of a slot transition from each word to once per slot and by utilizing information about which speaker is speaking in the slot transitions. In addition, work is underway to incorporate this model in the search engine to measure the recognition performance.

4.3.2. ILT–BASED DISCOURSE MODELING

Work is also underway to model the discourse by making predictions of subsequent ILTs based on the previous ones, using a connectionist implementation. The ILT generated by our LR parser is a language independent frame structure containing three main slots, *speech-act*, *sentence-type* and *semantic frame* along with a few other secondary slots. The speech-act refers to the action performed by the sentence, e.g., suggest, accept, and reject. The sentence type refers to the surface form of the sentence, e.g., statement, yes/no question, wh-question, directive. The semantic frame refers to main semantic content of the sentence, e.g., busy, free, out-of-town. Other slots in the ILT are *who*, which is the person referred to by the frame, *what*, a possible non-person object, and *when* and *topic*, a representation of any temporal component of the utterance.

The top level slots of the most recent ILT are encoded into a pattern of binary inputs. This information along with a bit indicating whether the next utterance comes from the same or different speaker is fed into a multi layer neural network utilizing the back-propagation learning algorithm. The input vectors are

Transcription: Actually the twenty sixth and the twenty seventh I'll be at a seminar all day.

```
((SPEECH-ACT *REJECT)
 (SENTENCE-TYPE *STATE)
 (FRAME *SCHEDULED)
 (WHO ((FRAME *I)))
 (TOPIC
     ((FRAME *TIME-LIST)
      (CONNECTIVE AND)
      (ITEMS
          (*MULTIPLE*
              ((FRAME *SIMPLE-TIME)
               (DAY 26))
              ((FRAME *SIMPLE-TIME)
               (DAY 27))))))
 ( WHAT  ((FRAME *SEMINAR)
          (SPECIFIER INDEFINITE)))
 (WHEN
     ((FRAME *SPECIAL-TIME)
      (SPECIFIER WHOLE)
      (NAME DAY)))
 ( ADVERB ACTUALLY))
```

Figure 6: Example for the ILT representation

associated with a representation of the subsequent ILT. In preliminary work, the network was trained on 24 dialogues of hand coded ILTs from the ESST database. The network learned some characteristics of discourse behavior, and is good at making some predictions of likely fillers for the speech-act and sentence-type slots of the subsequent ILT. The relative strength of the output units can be used to determine the relative probability of competing fillers for a particular slot.

There are drawbacks with this experiment that are easy to solve: Twenty-four dialogues is not sufficient for good modeling of discourse. Increasing data for training is easy and should yield improved results. In addition, interjections often disrupted the context of a sentence; for instance small utterances, such as *Well* between two full sentences interfere with the association of the ILTs for the two sentences. Using the previous content bearing ILT to predict the next ILT, rather than just the previous ILT, increases context, and should boost results. Additionally, experiments with network architecture are called for. With improved results, the predictions would be used to aid speech recognition by interpolating language models appropriate for sentences containing the predicted slots.

4.3.3. USING PLAN INFERENCE TO CONSTRAIN THE NEXT UTTERANCE

We are also implementing a more traditional approach to discourse modeling, using a plan inferencing system to model the discourse between the two speakers. We are using a tripartite model that distinguishes between the domain level, problem solving level, and the discourse level [1, 2]. This model takes ILTs for sentences as they are uttered and incorporates them into a plan tree. The plan tree indicates the function of each sentence at each plan level. For example, it will indicate

that the sentence *Are you free on Tuesday* is a yes-no question functioning as a suggestion at the discourse level. At the problem solving level, it is an attempt to instantiate a variable, namely the day of a meeting, and at the domain level it is part of a plan to have a meeting. The plan tree also shows how sentences are related to each other. For example, *I'm free at 2:00 p.m.* could function as an acceptance of the suggestion to meet on Tuesday. It could also be an implicit rejection of a suggestion to meet at one p.m., a suggestion itself, or simply background information, depending on the context in which it appears. The rule based discourse tracker identifies the correct speech-act for the utterance using domain knowledge from a temporal expert program and the current plan tree. Determining the correct speech-act can be vital for correct translation [18].

A focus stack identifies nodes of the plan tree to which children nodes can be attached. Nodes that are not in focus represent conversational segments that are closed or finished. The plan tree together with the focus stack constitute a discourse state. By modeling the discourse state, we can set constraints on what the next utterance might be. In order to use this information to constrain speech recognition, we will train several language models based on different discourse states. The speech recognizer will then interpolate the language models according to the current discourse state. Should the next utterance appear to be a non-sequitur, that is, not have an appropriate place to attach to the plan tree, we could attempt to rerecognize the utterance [19], or initiate an interactive repair module that would query the speaker in order to properly interpret the utterance [20].

5. CONCLUSION

Integration of speech and natural language processing is an important aspect in a multi-lingual speech translation system. Our work has focused on robust parsing of spoken language and using discourse knowledge to constrain the speech decoder. We have suggested statistical, knowledge-based and connectionist approaches to model the dialog structure. Although we are far from our ultimate goal of general-purpose accurate machine translation of spontaneous discourse, we have made significant initial strides. In particular, the inter-communicating modular design of JANUS, coupled with the complete separation of processing modules and data sources gives us a high degree of language and domain independence. The language or domain specific information is all encoded in data structures. This architecture enables us to enhance each module, as well as retain feedback and inter-module communication and integrated testing over time. We believe that such a cooperating modular architecture is the best way to address a problem as complex as translation of spontaneous spoken discourse.

6. REFERENCES

[1] L. Lambert, S. Carberry: *Modeling Negotiation Subdialogues,* 30th Annual Meeting of the Association for Computational Linguistics, 1992.

[2] L. Lambert: *Recognizing Complex Discourse Acts: A Tripartite Plan Based Model of Discourse,* PhD Dissertation, Department of Computer Science, University of Delaware Tech. Rep. 93-19, 1993.

[3] H. Kitano and C. Van Ess-Dykema: *Toward a Plan-Based Understanding Model for Mixed-Initiative Dialogues,* in Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL), Berkeley, 1991, 25-32.

[4] S. R. Young, A. G. Hauptmann, W. H. Ward, E. T. Smith, P. Werner: *High Level Knowledge Sources in Usable Speech Recognition Systems,* Communications of the ACM, Volume 32 Number 2, February 1989, p 183 - 194.

[5] S. R. Young: *Use of Dialog, Pragmatics and Semantics to Enhance Speech Recognition,* Speech Communication 9, 1990, pages 551-564.

[6] W. Ward, S. Young: *Flexible Use of Semantic Constraints in Speech Recognition,* IEEE International Conference on Acoustics, Speech and Signal Processing, Minneapolis, 1993, Vol. 2, pp. 49–50

[7] M. Woszczyna, N. Aoki-Waibel, F.D. Buo, N. Coccaro, K. Horiguchi, T. Kemp, A. Lavie, A. McNair, T. Polzin, I. Rogina, C.P. Rose, T. Schultz, B. Suhm, M. Tomita, A. Waibel: *JANUS 93: Towards Spontaneous Speech Translation,* to appear in ICASSP 94.

[8] H. Iida, T. Yamaoka, H. Arita: *Predicting the Next Utterance Linguistic Expressions Using Contextual Information,* in IEICE Trans. Inf. & Suyst., Vol. E76-D, No. 1, January 1993.

[9] S. Oviatt: *Predicting and Managing Spoken Disfluencies during Human-Computer Interaction,* to appear in Proceedings of the ARPA Human Language Technology workshop, Plainsboro, 1994

[10] A. Lavie, M. Tomita: *GLR* - An Efficient Noise-skipping Parsing Algorithm for Context-free Grammars,* Proceedings of Third International Workshop on Parsing Technologies, 1993, pp. 123–134

[11] A. Lavie: *An Integrated Heuristic for Partial Parse Evaluation,* to appear in Proceedings of 32nd Annual Meeting of the ACL, 1994

[12] T. Briscoe, J. Carroll: *Generalized Probabilistic LR Parsing of Natural Language (Corpora) with Unification-Based Grammars,* Computational Linguistics 1993, Vol. 19, pp. 25–59

[13] W. Ward: *Understanding Spontaneous Speech: The Phoenix System,* IEEE International Conference on Acoustics, Speech and Signal Processing, 1991, Vol. 1, pp. 365–367

[14] W. Ward: *Understanding Spontaneous Speech,* Proceedings of the DARPA Speech and Natural Language Workshop, 1989, pp 137-141

[15] W. Ward: *The CMU Air Travel Information Service: Understanding Spontaneous Speech,* Proceedings of the DARPA Speech and Natural Language Workshop, 1990

[16] R. Kneser, H. Ney: *Improved Clustering Techniques for Class-Based Statistical Language Models,* EUROSPEECH 93, Berlin, Vol. 2, pp. 973-976

[17] R. Pieraccini, E. Levin, C.-H. Lee: *Stochastic Representation of Conceptual Structure in the ATIS Task,* Proceedings of the DARPA Speech and Natural Language Workshop, 1992

[18] K. Kogura, M. Kume, H. Iida: *Illocutionary Act Based Translation of Dialogues,* The Third International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language, 1990

[19] S. Young, W. Ward: *Semantic and Pragmatically based Re-Recognition of Spontaneous Speech,* EUROSPEECH 93, Berlin, Vol. 3, pp. 2243-2247

[20] Carolyn Penstein Rosé, Alex Waibel: *Recovering From Parser Failures: A Hybrid Statistical/Symbolic Approach,* to appear in "The Balancing Act: Combining Symbolic and Statistical Approaches to Language" workshop at the 32nd Annual Meeting of the ACL, 1994