PARSING REAL INPUT IN JANUS: A CONCEPT-BASED APPROACH TO SPOKEN LANGUAGE TRANSLATION

L. Mayfield, M. Gavaldà, Y-H. Seo, B. Suhm, W. Ward, A. Waibel

Interactive Systems Laboratories
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213-3890
USA

ABSTRACT

As part of the JANUS speech-to-speech translation project[5], we have developed a translation system that successfully parses full utterances and is effective in parsing spontaneous speech, which is often syntactically ill-formed. The system is concept-based, meaning that it has no explicit notion of a sentence but rather views each input utterance as a potential sequence of concepts. Generation is performed by translating each of these concepts in whole phrases into the target language, consulting lookup tables only for low-level concepts such as numbers. Currently, we are working on an appointment scheduling task, parsing English, German, Spanish, and Korean input and producing output in those same languages and also Japanese.

1. INTRODUCTION

JANUS-2 [8] is a speech-to-speech translation system that translates spontaneous spoken input in English, German, Spanish, and Korean into English, German, Spanish, Korean, and Japanese. The translation component of this system must be able not only to handle the kinds of disfluencies that occur in normal speech but also to compensate for errors likely to occur during recognition.

In JANUS-1 [7], we used a syntactic parser that mapped input text onto interlingua text (ILT) representations which could then be used to generate a target-language translation. As we began to work with spoken input, however, we quickly found that the syntactic parser was not able to handle fragmented and "multi-sentence" utterances; moreover, spontaneous speech contains many more words than are actually necessary to communicate the speaker's intent and it was not clear that it was even desirable to translate them all.

We took a completely semantic approach to this problem. We are not translating in the traditional sense but rather producing an equivalent message in the target language based on meaning. Our parser is domain-limited but very robust; a tight semantic grammar within the scheduling domain captures possible topics without syntactic cues. The process is easily ported to other domains.

This approach is particularly well-suited to processing of spontaneous speech because of its robust handling of the particular phenomena of spoken input. One problem special to processing of spontaneous speech is that of fragmented and run-on sentences. Some systems require utterances to be hand-segmented before parsing. The techniques presented here, however, take advantage of the fact that syntactically ill-formed utterances are often *semantically* well-formed and breaks each input string into concept units (tokens) representing basic ideas such as time and availability. Grammatical constraints are introduced at the phrase level and regulate the semantic rather than the syntactic category. This method allows the ungrammaticalities that often occur between phrases to be ignored. We can thus handle complete turns, regardless of length or number of constituent concepts. An example of a spontaneous utterance, showing ungrammaticalities, is given in Figure 2 in Appendix C.

2. SYSTEM OVERVIEW

2.1. Recognition

The baseline JANUS-2 recognizer decodes speech in the source language into either a list of sentence candidates (Nbest) or a word lattice. As front end preprocessing, linear discriminant analysis (LDA) is used to find an optimal set of features, based on a Melscaled Fourier spectra and other acoustic features. After preprocessing, decoding is performed in two passes: first a Viterbi search as forward pass to find the first-best hypothesis, followed by a word-dependent backward pass to find an Nbest list or a word lattice. The three main knowledge sources for the decoder are a single pronunciation dictionary, continuous HMM tied on a phonetic level as acoustic models, and word bigram and trigram language models.

2.2. Parsing

There are two parsers associated with the JANUS project. The running real-time end-to-end system described in this paper uses a concept-based parser [5]. This parser produces a less detailed analysis but one that is possibly more robust when working with spoken language. GLR*, the LR parser described in [2, 3], constructs a language-independent interlingua text (ILT) representation of the input utterance and can produce a more precise parse with appropriate input, but also requires more detailed and complex grammars, and has greater computational requirements.

2.2.1. Concept Based Approach

The parsing module currently being used in JANUS-2 is an extension of the Phoenix Spoken Language System [11]. It tries to model the information structures in a scheduling task and the different ways these structures can be realized in words, identifying constituent concepts and matching segments of the input string to tokens. Although the words used to encode the concepts necessary to perform a given task differ, the set of concepts itself is language-independent, and we have developed a core set of approximately 120 tokens from 45 example English dialogues that is sufficient to model all of the input languages for the appointment scheduling task. All input languages are processed using the same technique.

Tokens represent all semantic categories from speech acts to individual variables such as numbers and days of the week. Examples of top level tokens, also called slots, in the scheduling frame would be giving of information and agreement or rejection. Intermediate tokens might differentiate between points and intervals of time, and bottom level tokens represent specific words that must be translated.

The parser may string together slots in any order. It is not always clear, however, where the slot boundaries should be. In these cases it follows a simple algorithm for determining how the utterance should be segmented. If there is no single interpretation which has the most words matched, the parser looks for the interpretation with the fewest number of slots. If there is more than one least fragmented interpretation, it picks the one with the largest number of nested tokens within the slots. This approach is described in more detail in [9].

This system is effect-oriented, meaning that the goal is to cause the listener to respond in the desired manner. Expressions that look very different are often mapped to the same token if they serve the same discourse function. For example, the utterances "what do you think" and "let me know" are both parsed as *your_turn*, indicating that the speaker is asking the listener for a reaction to what he has said. Figure 1 shows examples of this slot in the different input languages. The system does not recognize varying degrees of reluctance or desire, only basic acceptance and rejection. A finer-grained representation can be created by simply adding more tokens to reflect these nuances, and limiting the types of expressions that can be matched to each concept. We are exploring the possibility, however, of using the concept-based parser to produce a first-pass parse, and taking advantage of the ability of the GLR* parser to produce a precise parse when presented with appropriately segmented data. Phoenix could then be used also as a backup parser.

Developing each parsing grammar for this system took approximately three person-months. An example of the grammar specifications is given in Figure 6. Adding new structures to the grammar involves simply including new rewrite rules specifying the desired pattern. Generation grammar development took on the order of three person-weeks.

2.3. Generation

The generation component of the translator consists of a text processor and a translation grammar. As in the parsing grammar, the generation grammar contains one grammar file for each token. Grammar files for bottom-level tokens such as *day_of_week* and *month_name* are simple lookup tables. For all other tokens, the grammar file contains a list of templates which are target language phrases into which subtoken values can be inserted. The input parse is traced through left-to-right, and when lowest-level tokens are reached the correct target language value is extracted from the lookup table. The process then reverses, and these values are inserted into the parent phrase, which may itself in turn be inserted into a parent phrase. The process continues with sister tokens. This method works extremely well when translating between languages with similar morphological structure and word order. Figure 7 shows a sample generation grammar.

3. DIFFICULTIES IN PARSING SPOKEN DIALOGUES

Speech-to-speech translation differs from text-to-text translation in several important ways. Spontaneous speech contains human noise such as filler words (um, er) partial words and lip smacks.

It also often contains such phrase-level phenomena as mid-utterance corrections and bad word placement. Humans speaking naturally do not present their thoughts in clear and complete sentences; the idea of a sentence in spontaneous speech is unrealistic. Even read transcriptions of spontaneous speech do not duplicate the phenomena found in true spontaneous speech [4].

System components must not only be able to do their own jobs, but they must also be able to work with the kind of unpredictable input real-life systems face. Parsers that expect artificially segmented input are difficult to integrate with recognizers that produce a string of words devoid of syntactic markers in an end-to-end system. Many independently operational parsers can construct elegant representations of spontaneous speech presented in convenient units. If data must be manually modified between decoding and parsing, however, such a parser cannot be an effective part of a full system. Because our system's parser is able to process unbroken input, it can be incorporated in a fully functional end-to-end system.

Different approaches have been attempted to handle the particular problems of spontaneous speech. Syntactic systems work well for parsing text, but can be fragile when confronted with ill-formed input. It is possible to solve problems of coverage by placing restrictions on the words that a user may use or problems of segmentation by requiring him to speak in fixed units [6] but this is not realistic in a system that expects to act as an intermediary between humans speaking normally. Agnäs et al.[1] reported first-year end-to-end results for utterances with under 12 words in the ATIS task. In the scheduling dialogues, however, utterances average over 25 words in length in English and over 35 in Spanish. This kind of input *must* be either processed as is or segmented into units that the parser can handle automatically in an end-to-end system. The CMU GLR* parser can match input sentences to detailed ILTs *if* the sentences are in the proper format. To ensure that the parser will not fail, it has been necessary that input be manually checked and markers inserted *after* speech decoding, or such markers could be generated based on possible sentence breaks, such as pauses and after prosodic cues.

Time is also a consideration. To ensure natural human communication, fast response is crucial for automatic interpreting systems. The Phoenix parser averages 16 ms per utterance in SST.

Figure 3 shows a transcribed utterance with the markers necessary for LR parsing manually inserted. This utterance contains six sentences of the type expected by the parser and is typical of the data collected. If sentence boundaries had to be added mid-process full system integration is slowed.

4. EXPERIMENTS

4.1. Robust handling of full utterances

Figure 4 shows data as output by the speech recognizer. As mentioned earlier, the concept-based parser views each input utterance as a potential series of concepts. Utterances segmented using the method shown in Figure 3 are generally parsed as a single slot. Full utterances such as that in Figures 2 and 4 are simply longer, in this case a series of seven slots. Figure 5 shows how this utterance is parsed using our concept-based method.

In the integrated system, users see a paraphrase of their utterance in their own language before the translation is sent. This is created at the same time as foreign-language generations. This step ensures accurate translation, as the user can rephrase himself if he feels that the paraphrase is not accurate.

Target Language	Source languages - % coverage			
	English	German	Spanish	Korean
English	77.0 - 91.0	85.0 - 92.4	58.0 - 88.2	61.5 - 82.5 (E-K)

Table 1: Current performance ranges on translation into English.

4.2. System Evaluation

4.2.1. Procedure

The data used for development and testing of JANUS-2 was gathered following the conventions used by other sites in Europe, Japan and the U.S. working on the scheduling task. The same method was used to collect English, German, and Spanish data, ensuring consistency between languages as well as between dialogues. Participants were given one of 13 calendars marked with meetings, classes, and other commitments and asked to schedule a two-hour meeting. These dialogues were recorded and transcribed, using standard transcription and spelling conventions as shown in Figure 2. This method of data collection ensures that dialogues are natural and spontaneous yet limited in domain. For the evaluations, the systems were run on unseen tests set of approximately 100 turns.

4.2.2. Results

Table 1 shows generation results. Generation evaluation is necessarily very subjective. Native speakers of the target language who are fluent in the source language were asked to judge whether all of the important information in the source utterances is conveyed in the translations. Judges saw only the original transcribed utterances and the final translations.

Table 1 shows ranges of end-to-end coverage recently achieved. Clearly, performance depends greatly on the input dialogues. Higher numbers are from evaluation on dialogues in which only time of meeting was discussed; lower numbers are on dialogues in which locations (directions) and telephone calls, spellings, and availability of a third person were also discussed. Not all translation pairs have been evaluated; English to Korean translation results are shown here because the system has not yet been evaluated for Korean to English. The numbers shown here reflect evaluation of the full translation system on transcribed data.

5. CONCLUSION

The concept-based implementation of a spontaneous speech translator described in this paper is effective in an end-to-end system because of its speed, simplicity, and robust operation over spoken utterances. It allows straightforward handling of fragmented and multi-sentence utterances, processing them as easily as syntactically well-formed sentences. The system is easy to implement, and we have integrated this parser with speech recognition and synthesis modules in the JANUS-2 speech-to-speech translator. It should be able to provide incremental translation, and we hope to combine it with the GLR* parser, using the concept-based parser as a first pass and backup and GLR* to analyze them, to create a powerful parser that is both robust and precise.

A. ACKNOWLEDGEMENTS

This research was sponsored in part by the Department of the Navy, Office of Naval Research under Grant No. N00014-93-1-0806. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. We also gratefully acknowledge partial support by the Advanced Telecommunication Research Laboratories and by NEC Research Laboratories. Many thanks to Lori Levin and Sheryl Young, and especially Alex Waibel for advice and suggestions.

B. REFERENCES

[1] M-S Agnäs et al. Spoken Language Translator: First-Year Report. *SICS Research Report R94:03, SRI Cambridge Report CRC-043, ISSN 0283-3638*, 1994.

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

[3] L. Levin, O. Glickman, Y. Qu, D. Gates, A. Lavie, A. Waibel, C. Van Ess-Dykema. Using Context in Machine translation of Spoken Language. To appear in *Proceedings of TMI95*, 1995.

[4] P. Jeanrenaud, E. Eide, U. Choudhari, J. McDonough, K. Ng, M. Siu, H. Gish. Reducing Word Error Rate on Conversational Speech from the Switchboard Corpus *Proceedings of ICASSP-95*, 1995.

[5] L.J. Mayfield, M. Gavaldà, W. Ward, and A. Waibel. Concept-based Speech Translation. *Proceedings of ICASSP-95*, 1995.

[6] T. Morimoto, T. Takezawa, F. Yato, S. Sagayama, T. Tashiro, M. Nagata, A. Kurematsu. ATR's Speech Translation System: ASURA. In *Proceedings of Eurospeech '93* ,1993.

[7] L. Osterholtz, A. McNair, I. Rogina, H. Saito, T. Sloboda, J. Tebelskis, A. Waibel, and M. Woszczyna. Testing Generality in JANUS: A Multi-Lingual Speech to Speech Translation System In Proceedings of ICASSP-92, 1992.

[8] B. Suhm, L. Levin, N. Coccaro, J. Carbonell, K. Horiguchi, R. Isotani, A. Lavie, L. Mayfield, C.P. Rosé, C. Van Ess-Dykema, A. Waibel Speech-language Integration in a Multi-lingual Speech Translation System. In *Proceedings of the Workshop on Integration of Natural Language and Speech Processing, AAAI*, 1994.

[9] W. Ward. Understanding Spontaneous Speech. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 137, 141, 1989.

[10] W. Ward. The CMU Air Travel Information Service: Understanding Spontaneous Speech. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 127, 129, 1990.

[11] W. Ward. Extracting Information in Spontaneous Speech. In *Proceedings of International Conference on Spoken Language Processing*, 1994.

C. FIGURES

REALIZATIONS OF YOUR_TURN

Language	Mappings
English	[your_turn] (how's it look?) [your_turn] (do you have any ideas?)
German	[your_turn] (was meinen sie dazu?) [your_turn] (wie sieht es aus?)
Spanish	[your_turn] (?'qué te parece?) [your_turn] (?'le conviene?)

Figure 1: *Examples of phrases that are mapped to your_turn in the three input languages.*

A TYPICAL UTTERANCE

unfortunately i'll be out of town from the ninth through the eleventh
checking my calendar friday's no good either let's see maybe next week oh
that's bad my class schedule's okay how 'bout on tuesday the sixteenth any
time after twelve thirty

Figure 2: *Spontaneous utterance as transcribed by a human.*

/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@
{comma} {seos} > okay {comma} /h#/ how 'bout on Tuesday the sixteenth {comma}
any time after twelve thirty {period} #key_click# /h#/ /h#/ {seos}

Figure 3: *Sample transcription with markers for LR parser.* {*seos*} *marks the end of the semantic sentence unit.*

```
ON POSSIBLY I+LL BE OUT OF TOWN FROM THE NINTH THROUGH THE ELEVENTH LUNCH
AT LIKE HOW ONE THIRTY AND FRIDAY+S NO GOOD AT I DAY THERE+S  THE SEE MAYBE
NEXT WEEK AND NEXT THEN SCHEDULES OKAY HOW +BOUT ON TUESDAY THE SIXTEENTH
ANYTIME AFTER TWELVE THIRTY
```

Figure 4: Spontaneous utterance as decoded by the recognizer.

```
-UNFORTUNATELY i+ll be out of town from the ninth through the eleventh
-CHECKING *MY *CALENDAR friday+s no good either let+s see maybe next week
*OH that+s bad *MY *CLASS -SCHEDULE+S *OKAY how +bout on tuesday the
sixteenth any time after twelve thirty -#CLICK#

Interpretation score 32
Frame scheduling    score= 32    num_slots= 7

[give_info] ( [my_unavailability] ( I+LL BE [out_of_town] ( OUT OF TOWN
     [temporal] ( [interval] ( FROM [start_point] ( [date] ( THE
          [day_ord] ( NINTH )))THROUGH [end_point] ( [date] ( THE
          [day_ord] ( ELEVENTH ))))))))
[give_info] ( [my_unavailability] ( [temporal] ( [point] (
          [d_o_w] ( FRIDAY+S )))NO GOOD ))
[interject] ( [conj] ( EITHER ))
[interject] ( LET+S SEE )
[temporal] ( [point] ( MAYBE [next_week] ( NEXT WEEK )))
[give_info] ( [my_unavailability] ( [anaphoric] ( THAT+S )BAD ))
[suggest_time] ( HOW +BOUT [temporal] ( [point] ( ON [date] (
     [d_o_w] ( TUESDAY )THE [day_num] ( SIXTEENTH )))
     [range] ( ANY TIME [after] ( AFTER )[time] (
          [hour] ( TWELVE )[minute] ( THIRTY )))))
```

Figure 5: Concept-based parse of the utterance in Figure 2. Skipped words are shown in capitals in the interpretation at the top; those marked with (-) are out-of-lexicon and those marked with (*) are known to the system but unexpected in this environment.

```
[my_unavailability]
        (i *BABBLE CANT *MEET +[temporal])
        (+[temporal] BE *BABBLE BAD *FOR_ME)

BABBLE                          BE
        (really)                        (is)
        (probably)                      (would be)
        (kind of)
        (unfortunately)         BAD
                                        (bad)
CANT                                    (tight)
        (can't)                         (booked solid)
        (couldn't)                      (packed)
        (don't want to)                 (out)
                                        (no good)
MEET
        (meet)                  FOR_ME
        (do it)                         (for me)
        (make it)                       (here)
```

Figure 6: Sample grammar file for [my_unavailability]. Words marked with (*) are optional; words marked with (+) may repeat. Capitalized words that appear in a rewrite rule are nonterminal and expansions are shown below. Lower-case words are terminal; words in brackets are token names and are represented by separate grammar files. Example patterns that this network would match are *i really couldn't do it next week and the week after* and *Tuesday would be kind of tight.*

```
I'm busy [temporal].
```

Figure 7: Generation grammar for the token [my_unavailability] covering the patterns in Fig. 6. The example sentences in Fig. 6 would then be translated as *I'm busy next week and the week after* and *I'm busy Tuesday.*