Multi-lingual Translation of Spontaneously Spoken Language in a Limited Domain

Alon Lavie, Donna Gates, Marsal Gavaldà, Laura Mayfield, Alex Waibel and Lori Levin
Center for Machine Translation
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
e mail : lavie@cs.cmu.edu

Abstract
JANUS is a multi-lingual speech-to-speech translation system designed to facilitate communication between two parties engaged in a spontaneous conversation in a limited domain. In an attempt to achieve both robustness and translation accuracy we use two different translation components: the GLR module, designed to be more accurate, and the Phoenix module, designed to be more robust. We analyze the strengths and weaknesses of each of the approaches and describe our work on combining them. Another recent focus has been on developing a detailed end-to-end evaluation procedure to measure the performance and effectiveness of the system. We present this procedure in the latter part of the paper, and present our most recent Spanish-to-English performance evaluation results.

1 Introduction
JANUS is a multi-lingual speech-to-speech translation system designed to facilitate communication between two parties engaged in a spontaneous conversation in a limited domain. In this paper we describe the current design and performance of the machine translation module of our system. The analysis of spontaneous speech requires dealing with problems such as speech disfluencies, looser notions of grammaticality and the lack of clearly marked sentence boundaries. These problems are further exacerbated by errors of the speech recognizer. We describe how our machine translation system is designed to effectively handle these and other problems. In an attempt to achieve both robustness and translation accuracy we use two different translation components: the GLR module, designed to be more accurate, and the Phoenix module, designed to be more robust. Both modules follow an interlingua-based approach. The translation modules are designed to be language-independent in the sense that they each consist of a general processor that applies independently specified knowledge about different languages. This facilitates the easy adaptation of the system to new languages and domains. We analyze the strengths and weaknesses of each of the translation approaches and describe our work on combining them. Our current system is designed to translate spontaneous dialogues in the Scheduling domain, with English, Spanish and German as both source and target languages. A recent focus has been on developing a detailed end-to-end evaluation procedure to measure the performance and effectiveness of the system. We describe this procedure in the latter part of the paper, and present our most recent Spanish-to-English performance evaluation results.

2 System Overview
The JANUS System is a large scale multi-lingual speech-to-speech translation system designed to facilitate communication between two parties engaged in a spontaneous conversation in a limited domain. A diagram of the architecture of the system is shown in Figure 1. The system is composed of three main components: a speech recognizer, a machine translation (MT) module and a speech synthesis module. The speech recognition component of the system is described elsewhere (Woszczyyna et al. 1994). For speech synthesis, we use a commercially available speech synthesizer.

The MT module is composed of two separate translation sub-modules which operate independently. The first is the GLR module, designed to be more accurate. The second is the Phoenix module, designed to be more robust. Both modules follow an interlingua-based approach. The source language input string is first analyzed by a parser, which produces a language-independent interlingua content representation. The interlingua is then passed to a generation component, which produces an output string in the target language.

The discourse processor is a component of the GLR translation module. It disambiguates the speech act of each sentence, normalizes temporal
expressions, and incorporates the sentence into a discourse plan tree. The discourse processor also updates a calendar which keeps track of what the speakers have said about their schedules. The discourse processor is described in greater detail elsewhere (Rosé et al. 1995).

3 The GLR Translation Module

The GLR* parser (Lavie and Tomita 1993; Lavie 1994) is a parsing system based on Tomita’s Generalized LR parsing algorithm (Tomita 1987). The parser 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. This is done using a beam search heuristic that limits the combinations of skipped words considered by the parser, and ensures that it operates within feasible time and space bounds.

The GLR* parser was implemented as an extension to the GLR parsing system, a unification-based practical natural language system (Tomita 1990). The grammars we develop for the JANUS system 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. The main components 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, busy). An example of an ILT is shown in Figure 2. A detailed ILT Specification was designed as a formal description of the allowable ILTs. All parser output must conform to this ILT Specification. The GLR unification based formalism allows the grammars to construct precise and very detailed ILTs. This in turn allows the GLR translation module to produce highly accurate translations for well-formed input.

The GLR* parser also includes several tools designed to address the difficulties of parsing spontaneous speech. To cope with high levels of ambiguity, the parser includes a statistical disambiguation module, in which probabilities are attached directly to the actions in the LR parsing table. The parser can identify sentence boundaries within each hypothesis with the help of a statistical method that determines the probability of a boundary at each point in the utterance. The parser must also determine the “best” parse from among the different parseable subsets of an input. This is done using a collection of parse evaluation measures which are combined into an integrated heuristic for evaluating and ranking the parses produced by the parser. Additionally, a parse quality heuristic allows the parser to self-
judge the quality of the parse chosen as best, and to detect cases in which important information is likely to have been skipped.

Target language generation in the GLR module is done using GenKit (Tomita and Nyberg 1988), a unification-based generation system. With well-developed generation grammars, GenKit results in very accurate translation for well-specified ILTs.

4 The Phoenix Translation Module

The JANUS Phoenix translation module (Mayfield et al. 1995) is an extension of the Phoenix Spoken Language System (Ward 1991; Ward 1994). The translation component consists of a parsing module and a generation module. Translation between any of the four source languages (English, German, Spanish, Korean) and five target languages (English, German, Spanish, Korean, Japanese) is possible, although we currently focus only on a few of these language pairs.

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. Whereas GLR is general enough to support both semantic and syntactic grammars (or some combination of both types), the Phoenix approach was specifically designed for semantic grammars. Grammatical constraints are introduced at the phrase level (as opposed to the sentence level) and regulate semantic categories. This 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.

The parsing grammar specifies patterns which represent concepts in the domain. The patterns are composed of words of the input string as well as other tokens for constituent concepts. Elements (words or tokens) in a pattern may be specified as optional or repeating (as in a Kleene star mechanism). Each concept, irrespective of its level in the hierarchy, is represented by a separate grammar file. These grammars are coupled into Recursive Transition Networks (RTNs).

The interlingua meaning representation of an input utterance is derived directly from the parse tree constructed by the parser, by extracting the represented structure of concepts. This representation is usually less detailed than the corresponding GLR ILT representation, and thus often results in a somewhat less accurate translation. The set of semantic concept tokens for the Scheduling domain was initially developed from a set of 45 example English dialogues. Top-level concepts, also called slots, represent speech acts, such as suggestion or agreement. Intermediate-level tokens distinguish between points and intervals in time, for example; lower-level tokens capture the specifics of the utterance, such as days of the week, and represent the only words that are translated directly via lookup tables.

The parser matches as much of the input utterance as it can to the patterns specified by the RTNs. Out-of-lexicon words are ignored, unless they occur in specific locations where open concepts are permitted. A word that is already known to the system, however, can cause a concept pattern not to match if it occurs in a position unspecified in the grammar. A failed concept does not cause the entire parse to fail. 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 can cause added 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.

Figure 3 shows an example of a speaker utterance and the parse that was produced using the Phoenix parser. The parsed speech recognizer output is shown with unknown (-) and unexpected (+) words marked. These segments of the input were ignored by the parser. The relevant concepts, however, are extracted, and string together they provide a general meaning representation of what the speaker actually said.

Generation in the Phoenix module is accomplished using a simple strategy that sequentially generates target language text for each of the top level concepts in the parse analysis. Each concept has one or more fixed phrasings in the target language. Variables such as times and dates are extracted from the parse analysis and translated directly. The result is a meaningful translation, but can have a telegraphic feel.

5 Combining the GLR and Phoenix Translation Modules

5.1 Strengths and Weaknesses of the Approaches

As already described, both the GLR* parser and the Phoenix parser were specifically designed to handle the problems associated with analyzing spontaneous speech. However, each of the ap-

{(frame *free)
 |who {frame *}))
 |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 2: An Example ILT
Original utterance:
SI QUE TE PARECE TENG0 EL MARTES DIECIOCHO Y EL MIERCOLES DIECINUEVE LIBRES TODO EL DIA PODRIAMOS IR DE MATEINE O SEA EN LA TARDE VER EL LA PELICULA.
(Roughly: "Yes what do you think? I could meet Tuesday the eighteenth and Wednesday the nineteenth free all day we could go see the movie so in the afternoon see the the movie.")

As decoded by the recognizer:
%N0ISE% SI QUE TE PARECE %N0ISE% TENG0 EL MARTES DIECIOCHO Y EL MIERCOLES DIECINUEVE LIBRES TODO EL DIA PODRIAMOS IR DE MATEINE %N0ISE% O SEA LA TARDE A VER LA

Parsed:
%<CS> si quel te parece tengo el martes dieciocho y el miercoles diecinueve libres todo el dia podriamos *IR *DE -MATEINE o sea la tarde a ver LA %</CS>

Parse Tree (= Semantic Representation):
[respond] ( [pre] ( SI ))
[your turn] ( QUE1 TE PAEECE )
[give info] ( [my availability] ( TENGO [temp/loc] )
 ( [temporal] ( [point] ( [date] ( [EL [d`o`w] ( MARTES ) )
 ( [date] ( [day ord] ( DIECIOCHO ) [day] ( Y ) [EL [d`o`w] ( MIERCOLES ) [date] ( [day ord] ( DIECINUEVE ) )))
 ( [libres] )))
(suggest) ( [suggest meeting] ( [temp/loc] ( [temporal] ( [entire] ( TODO ) [EL [d`o`w] ( [d`o`w] ( [DE (dia) ])))PODRIAMOS ))
[point] ( [OUT] )

Generated:
English = "Yes what do you think? I could meet Tuesday the eighteenth and Wednesday the nineteenth free all day we could go see the movie so in the afternoon see the the movie."
7 Conclusions and Future Work

In this paper we described the design of the two translation modules used in the JANUS system, outlined their strengths and weaknesses and described our efforts to combine the two approaches. A newly developed end-to-end evaluation procedure allows us to assess the overall performance of the system using each of the translations methods separately or both combined.

Our evaluations have confirmed that the GLR approach provides more accurate translations, while the Phoenix approach is more robust. Combining the two approaches using the parse quality judgement of the GLR* parser results in improved performance. We are currently investigating other methods for combining the two translation approaches. Since GLR* performs much better when long utterances are broken into sentences or sub-utterances which are parsed separately, we are looking into the possibility of using Phoenix to detect such boundaries. We are also developing a parse quality heuristic for the Phoenix parser using statistical and other methods.

Another active research topic is the automatic detection of out-of-domain segments and utterances. Our experience has indicated that a large proportion of bad translations arise from the translation of small parsable fragments within out-of-domain phrases. Several approaches are under consideration. For the Phoenix parser, we have implemented a simple method that looks for small islands of parsed words among non-parsed words and rejects them. On a recent test set, we achieved a 33% detection rate of out-of-domain parses with no false alarms. Another approach we are pursuing is to use word salience measures to identify and reject out-of-domain segments.

We are also working on tightening the coupling of the speech recognition and translation modules of our system. We are developing lattice parsing versions of both the GLR* and Phoenix parsers, so that multiple speech hypotheses can be efficiently analyzed in parallel, in search of an interpretation that is most likely to be correct.

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.

We would like to thank all members of the JANUS teams at the University of Karlsruhe and Carnegie Mellon University for their dedicated work on our many evaluations.

---

2In a more recent evaluation, this combination method resulted in a 9.5% improvement in acceptable translations of speech recognized in-domain sentences. Although some variation between test sets is to be expected, this result strengthens our belief in the potential of this approach.
In Domain (605 sentences)

<table>
<thead>
<tr>
<th></th>
<th>GLR*</th>
<th>Phoenix</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>transcribed</td>
<td>speech</td>
</tr>
<tr>
<td>Perfect</td>
<td>65.2</td>
<td>34.7</td>
</tr>
<tr>
<td>OK</td>
<td>18.8</td>
<td>12.2</td>
</tr>
<tr>
<td>Bad</td>
<td>16.0</td>
<td>29.2</td>
</tr>
<tr>
<td>Recog Err</td>
<td>**</td>
<td>23.9</td>
</tr>
</tbody>
</table>

Out of Domain (485 sentences)

<table>
<thead>
<tr>
<th></th>
<th>GLR*</th>
<th>Phoenix</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>transcribed</td>
<td>speech</td>
</tr>
<tr>
<td>Perfect</td>
<td>58.5</td>
<td>29.7</td>
</tr>
<tr>
<td>OK</td>
<td>26.7</td>
<td>42.4</td>
</tr>
<tr>
<td>Bad</td>
<td>14.8</td>
<td>7.5</td>
</tr>
<tr>
<td>Recog Err</td>
<td>**</td>
<td>20.4</td>
</tr>
</tbody>
</table>

Acceptable (Perfect + OK)

<table>
<thead>
<tr>
<th></th>
<th>In Dom</th>
<th>Out of Dom</th>
<th>All Dom</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>84.0</td>
<td>85.2</td>
<td>84.5</td>
</tr>
<tr>
<td></td>
<td>46.9</td>
<td>72.1</td>
<td>58.2</td>
</tr>
<tr>
<td></td>
<td>78.6</td>
<td>88.8</td>
<td>82.9</td>
</tr>
<tr>
<td></td>
<td>61.8</td>
<td>70.4</td>
<td>65.5</td>
</tr>
</tbody>
</table>

Figure 4: September 1995 evaluation of GLR* and Phoenix. Cross-grading of 16 dialogues.

In Domain (605 sentences)

<table>
<thead>
<tr>
<th></th>
<th>GLR* with Phoenix</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>transcribed</td>
</tr>
<tr>
<td>Perfect</td>
<td>65.4</td>
</tr>
<tr>
<td>OK</td>
<td>20.8</td>
</tr>
<tr>
<td>Bad</td>
<td>13.8</td>
</tr>
<tr>
<td>Recog Err</td>
<td>**</td>
</tr>
</tbody>
</table>

Acceptable (Perfect + OK)

<table>
<thead>
<tr>
<th></th>
<th>In Dom</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>86.2</td>
</tr>
<tr>
<td></td>
<td>60.9</td>
</tr>
</tbody>
</table>

Figure 5: September 1995 evaluation of GLR* combined with Phoenix. Cross-grading of 16 dialogues.

References


