In Proc. of ECAI-96 Wrkshp on Dial. Proc. of SL Sys.

Input Segmentation of Spontaneous Speech in JANUS: a Speech-to-speech Translation System

Alon Lavie[1]**, Donna Gates, Noah Coccaro and Lori Levin**

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 this paper we describe how multi-level segmentation of single utterance turns improves translation quality and facilitates accurate translation in our system. We define the basic dialogue units that are handled by our system, and discuss the cues and methods employed by the system in segmenting the input utterance into such units. Utterance segmentation in our system is performed in a multi-level incremental fashion, partly prior and partly during analysis by the parser. The segmentation relies on a combination of acoustic, lexical, semantic and statistical knowledge sources, which are described in detail in the paper. We also discuss how our system is designed to disambiguate among alternative possible input segmentations.

Keywords: Speech Translation, Segmentation, Dialogue Units, Spontaneous Speech.

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. It currently translates spoken conversations in which two people are scheduling a meeting with each other. 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. In this paper we describe how multi-level segmentation of single utterance turns improves translation quality and facilitates accurate translation in our system. We define the basic dialogue units that are handled by our system, and discuss the cues and methods employed by the system in segmenting the input utterance into such units. Utterance segmentation in our system is performed in a multi-level incremental fashion, partly prior to and partly during analysis by the parser. The segmentation relies on a combination of acoustic, lexical, semantic and statistical knowledge sources, which are described in detail in the paper. We also discuss how our system is designed to disambiguate among alternative possible input segmentations.

The remainder of this paper is organized in the following way. We begin with an overview of the translation part of the JANUS system in Section 2. In Section 3 we define the basic dialogue units which we model in our system, and describe how our system goes about translating such basic units. Section 4 discusses our initial input segmentation that is performed prior to parsing. Section 5 deals with parse-time segmentation of the input into basic dialogue units, and addresses the issue of disambiguation among alternative segmentations. In Section 6 we report our most recent results from an end-to-end translation evaluation with and without pre-segmented input. Finally, we present our summary and conclusions in Section 7.

2 System Overview

A diagram of the general architecture of the JANUS system is shown in Figure 1. The JANUS 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 [8]. For speech synthesis, we use a commercially available speech synthesizer.

At the core of the system are two separate translation modules which operate independently. The first is the Generalized LR (GLR) module, designed to be more accurate. The second is the Phoenix module [5], designed to be more robust. Both modules follow an interlingua based approach. In this paper, we focus on the GLR translation module. The results that will be reported in this paper will be based on the performance of the GLR module except where otherwise noted.

The source language input string is first analyzed by the GLR* parser [2][3]. Lexical analysis is provided by a morphological analyzer [4] based on Left Associative Morphology [1]. The parser uses a set of grammar rules in a unification-based formalism to produce a language-independent interlingua content representation in the form of a feature structure. The parser is designed to be robust over spontaneous speech in that it skips parts of the utterance that it cannot incorporate into a well-formed interlingua. After parsing, the interlingua is augmented and completed by the discourse processor [6] where it is also assigned a speech-act, and then passed to a generation component [7], which produces an output string in the target language.

3 Semantic Dialogue Units for Speech Translation

Speech Translation in the JANUS system is guided by the general principle that spoken utterances can be analyzed and translated as a sequential collection of semantic dialogue units (SDUs), each of which roughly corresponds to a speech-act. SDUs are semantically coherent pieces of information. Each dialogue unit in an utterance can in principle be analyzed and translated independently, although context from previous dialogue units can often help in resolving ambiguity and producing more accurate translations. The interlingua representation in our system was designed to capture meaning at the level of such SDUs. Each semantic dialogue unit is analyzed into an interlingua representation.

[1] Center for Machine Translation, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213 (USA), email: lavie@cs.cmu.edu

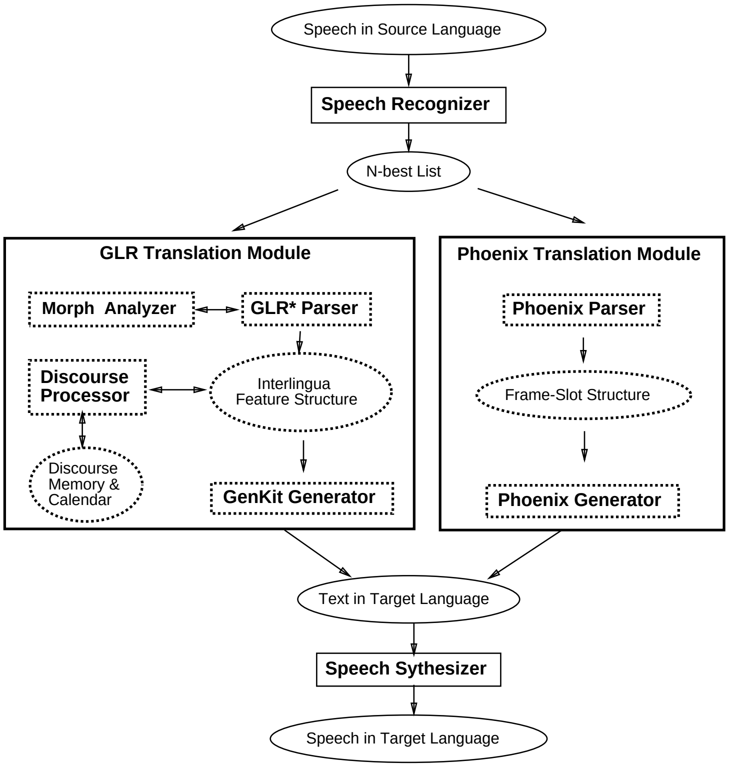

Figure 1. **The JANUS System**

```
mmxp_22_06: /h#/ si’ {period} {seos} mira {period} {seos}
toda la man˜ana estoy disponible {period} /h#/ {seos}
/eh/ y tambie’n los fin de semana {period} {seos}
si podri’a ser mejor un di’a de fin de semana {comma}
porque justo el once no puedo {period} {seos}
me es imposible {period} {seos} /gl/ {begin_simultaneous}
vos {end_simultaneous} pode’s el fin de semana {quest} {seos}
```

Figure 2. Transcription of An Utterance

3.1 Transcribing SDU boundaries

The dialogues that we use for development and testing are recordings of native speakers where two speakers have the task of scheduling a meeting. These recordings are transcribed with SDU boundaries indicated. While listening to the dialogues, the transcribers use acoustic signals as well as their own intuition of where sentential or fragment boundaries occur. Figure 2 shows an example of a transcribed utterance. The SDU boundaries are indicated with the transcription convention marking: *seos* (i.e., semantic end of sentence).

3.2 Parsing SDUs

Our analysis grammars are designed to produce analyses for the variety of naturally occuring SDUs in spontaneously spoken dialogues in the scheduling domain. More often than not, SDUs do not correspond to grammatically complete sentences. SDUs often correspond to input fragments, clauses or phrases, which convey a semantically coherent piece of information. In particular, fragments such as time expressions often appear in isolation, and are allowed to form complete analyses and "float" to the top level of our grammars.

```
Transcription of Segmented Utterance:

 (si1) (mira) (toda la man5ana estoy disponible)
 (y tambie1n los fin de semana)
 (si podri1a ser  mejor un di1a de fin
  de semana porque justo el once no puedo)
 (me es imposible)
 (vos pode1s el fin de semana)

Handmade Translation of the Utterance:

 Yes. Look, all morning I’m free.
 And also the weekend.  If it would be
 better, a day on the weekend, because on
 the eleventh I can’t (meet). It is impossible
 for me.  Can you (meet) on the weekend?
```

Figure 3. Semantic Dialogue Units in a Spanish Full Utterance

A full utterance turn thus consists of a sequence of SDUs. Figure 3 contains an example utterance in Spanish which demonstrates how a full utterance can consist of multiple SDUs. The analysis of a full utterance turn as a collection of semantic dialogue units requires the ability to correctly identify the boundaries between the units. This turns out to be a difficult yet crucial task, since translation accuracy greatly depends on correct segmentation. Utterance segmentation in our system is performed in a multi-level incremental fashion, partly prior to and partly during analysis by the parser. The segmentation relies on a combination of acoustic, lexical, semantic and statistical knowledge sources.

4 Pre-parsing Segmentation

Segmentation decisions in our system can be made reliably during parsing, at which point multiple sources of knowledge can be applied. Nevertheless, we have discovered that there are several advantages to be gained from performing some amount of segmentation at the pre-parsing stage. The goal here is to detect where SDU boundaries are highly likely. We then pre-break the utterance at these points into sub-utterances that may still contain multiple SDUs. Each of these sub-utterances is then parsed separately.

Pre-parsing segmentation at SDU boundaries is determined using acoustic, statistical and lexical information. The segmentation at the pre-parsing stage has two main advantages. The first advantage is a potential increase in the parsability of the utterance. Although the GLR* parser is designed to skip over parts of the utterance that it cannot incorporate into a meaningful structure, only limited amounts of skipping are considered by the parser, due to feasibility constraints. Often, a long utterance contains a long internal segment that is unparsable. If the parser does not manage to skip over the entire unparsable segment, a partial parse may be produced that covers either the portion of the utterance that preceded the unparsable segment, or the portion of the utterance that followed it, but not both. If, however, the utterance is pre-broken into several sub-utterances that are then parsed separately, there is a greater chance that parses will be produced for all portions of the original utterance.

The other potential advantage of pre-breaking an utterance is a significant reduction in ambiguity and subsequently a significant increase in system efficiency. Considering all possible segmentations of the utterance into SDUs adds an enormous level of ambiguity to the parsing process. A long utterance may in fact have hundreds of different ways by which it can be segmented into analyzable SDUs. This amount of ambiguity can be drastically reduced by determining several highly confident SDU boundaries in advance. Each of the sub-utterances passed on to the parser is then much smaller, and has far fewer possible segmentations into SDUs that must be considered by the parser. The example in Figure 4 demonstrates this. Without pre-breaking, the complete utterance is parsable, but requires 113 seconds. Pre-breaking produces the set of three sub-utterances shown in the middle of the figure. Parsing the three sub-utterances in sequence requires only 32 seconds (less than a third of the time).

4.1 Acoustic Cues for SDU Boundaries

The first source of information for our pre-breaking procedure is acoustic information supplied by the speech recognizer. We find that some acoustic cues have a very high probability of occurring at SDU Boundaries. These are dependent on the language to a certain extent. Long silences in general are a good indicator of an SDU boundary. After testing various combinations of noises within Spanish dialogues, we find that the following acoustic signals yield the best results for picking SDU boundaries: silences, two or more human noises in a row and three or more human or non-human noises in a row. However, since multiple noises and silences inside an utterance are rare in the Spanish data, pre-breaking according to acoustic cues still results most of the time in long unbroken utterances with which we must contend.

4.2 Statistical Detection of SDU Boundaries

The second source of information for our pre-breaking procedure is a statistically trained confidence measure that attempts to capture the likelihood of an SDU boundary between any pair of words in the utterance. The likelihood of a boundary at a particular point in the utterance is estimated based on a window of four words surrounding the potential boundary location (the two words prior to the point in question and the two words following it). 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. There are three bigram frequencies that are relevant to the decision of whether or not an SDU boundary is likely at this point. These are:

```
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 %noise%)

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)))

Actual 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 4. Efficiency Effect of Pre-breaking on a Spanish Full Utterance

1. $F([w_1 w_2 \bullet])$: the frequency of a clause boundary being to the right of the bigram $[w_1 w_2]$.
2. $F([w_2 \bullet w_3])$: the frequency of a clause boundary being in between the bigram $[w_2 w_3]$.
3. $F([\bullet w_3 w_4])$: the frequency of a clause boundary being to the left of the bigram $[w_3 w_4]$.

The bigram frequencies are estimated from a training set in which SDU boundaries are explicitly marked. The frequencies are calculated from the number of times an SDU boundary appeared in the training data in conjunction with the appropriate bigrams. In other words, if $C([w_i w_j \bullet])$ is the number of times that a clause boundary appears to the right of the bigram $[w_i w_j]$ and $C([w_i w_j])$ is the total number of times that the bigram $[w_i w_j]$ appears in the training set, then

$$F([w_i w_j \bullet]) = \frac{C([w_i w_j \bullet])}{C([w_i w_j])}$$

$F([w_i \bullet w_j])$ and $F([\bullet w_i w_j])$ are calculated in a similar fashion. However, for a given quadruple $[w_1 w_2 \bullet w_3 w_4]$, in order to determine whether the point in question is a reasonable place for breaking the utterance, we compute the following estimated frequency $\tilde{F}([w_1 w_2 \bullet w_3 w_4])$:

$$\tilde{F}([w_1 w_2 \bullet w_3 w_4]) = \frac{C([w_1 w_2 \bullet]) + C([w_2 \bullet w_3]) + C([\bullet w_3 w_4])}{C([w_1 w_2]) + C([w_2 w_3]) + C([w_3 w_4])}$$

This was shown to be more effective than the linear combination of the frequencies $F([w_1w_2\bullet])$, $F([w_2 \bullet w_3])$ and $F([\bullet w_3w_4])$. The method we use is more effective because a bigram with a low frequency of appearance, for which we may not have sufficiently reliable information, is not counted as highly as the other factors.

If the calculated SDU boundary probability $\tilde{F}([w_1w_2 \bullet w_3w_4])$ exceeds a pre-determined threshold, the utterance will be segmented at this point. Setting the threshold for segmentation too low will result in higher levels of segmentation, some of which will likely be incorrect. Setting the threshold too high will result in ineffective segmentation. As already mentioned, even though pre-parsing segmentation can improve system efficiency and accuracy, segmentation decisions in our system can be done much more reliably at the parser level. Furthermore, an incorrect pre-parsing segmentation cannot be corrected at a later stage. For these reasons, we set the threshold for pre-parsing segmentation to a cautiously high value, so as to prevent incorrect segmentations as much as possible. The actual threshold was determined based on experimentation with several values over a large development set of dialogues. We determined the lowest possible threshold value that still completely avoided incorrect segmentations. The statistical segmentation predictions were compared against the SDU boundary markers in the transcribed versions of the utterances to determine if a prediction was correct or false. The best threshold for pre-breaking was determined to be 0.6.

4.3 Lexical Cues for SDU Boundaries

The third source of information for our pre-breaking procedure is a set of lexical cues for SDU boundaries. These cues are language- and most likely 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. While examining the results of the statistical pre-breaking, we noticed that there were phrases that almost always occurred at SDU boundaries. These phrases usually had high SDU boundary probabilities, but in some cases not high enough to exceed the threshold for SDU boundary prediction. We examined roughly 100 dialogues from the Scheduling domain to find phrases that commonly occur at SDU boundaries. We determined that the phrases *qué tal*, *qué te parece*, and *si* usually occur after an SDU boundary while *Sí* and *claro* occur before an SDU boundary.

We modified the procedure for pre-breaking so that it could take into account these phrases. A small knowledge base of the phrases was created. The phrases alone do not trigger SDU boundary breaks. They are combined with the statistical component mentioned in the previous subsection. This is done by assigning the phrases a probability "boost" value. When the phrases are encountered by the pre-breaking program, the corresponding SDU boundary probability is incremented by the boost value of the phrase. After the optimal pre-breaking statistical threshold was determined to be 0.6, we experimented with several probability boost values for the phrases. These numbers were determined in the same manner as the best pre-breaking threshold. For phrases that occur after a boundary, we determined that the best probability boost value is 0.15. For phrases that occur before a boundary, the best probability boost value is 0.25. When one of the lexical phrases appears in an utterance, the increase in probability due to the boost value will usually increment the SDU boundary probability enough to exceed the threshold of 0.6. In cases where the probability still does not exceed the threshold, the original boundary probability is so low that it would be dangerous to break the utterance at that point. In such cases, we prefer to allow the parser to determine whether or not the point in question is in fact an SDU boundary.

5 Parse-time Segmentation and Disambiguation

Once the input utterance has been broken into chunks by our pre-parsing segmentation procedure, it is sent to the parser for analysis. Each utterance chunk corresponds to one or more SDUs. The GLR* parser analyzes each chunk separately, and must find the best way to segment each chunk into individual SDUs. Chunks that contain multiple SDUs can often be segmented in several different ways. As mentioned in the previous section, the number of possible SDU segmentations of a chunk greatly increases as a function of its length. This presents the parser with a significant additional level of ambiguity.

Even single SDUs may often have multiple analyses according to the grammar, and may thus prove to be ambiguous. The level of ambiguity introduced by chunks of multiple SDUs can drastically compound this problem. Dealing with such high levels of ambiguity is problematic from two different perspectives. The first is parser efficiency, which is directly correlated to the number of different analyses that must be considered in the course of parsing an input. The second perspective is the accuracy of the selected parse result. The greater the amount of ambiguity, the more difficult it is for the parser to apply its disambiguation methods successfully, so that the most "correct" analysis is chosen. The task of finding the "best" segmentation is therefore an integral part of the larger parse disambiguation process.

The early pruning of ambiguities that correspond to chunk segmentations that are unlikely to be correct can result in a dramatic reduction in the level of ambiguity facing the parser. This can result in a significant improvement in both parser efficiency and accuracy.

5.1 The Fragmentation Counter Feature

Because our grammar is designed to be able to analyze fragments as first class SDUs, it is often the case that an input chunk can be analyzed both as a single SDU, as well as a sequence of smaller fragment SDUs. In most cases, when such a choice exists, the least fragmented analysis corresponds to the most semantically coherent representation. We therefore developed a mechanism for representing the amount of fragmentation in an analysis, so that less fragmented analyses could be easily identified.

The fragmentation of an analysis is reflected via a special "counter" slot in the output of the parser. The value of the counter slot is determined by explicit settings in the various rules of the grammar. This is done by unification equations in the grammar rule that set the value of the counter slot in the feature structure that corresponds to the left-hand side non-terminal of the grammar rule. In this way, the counter slot can either be set to some desired value, or assigned a value that is a function of counter slot values of constituents on the right-hand side of the rule.

By assigning counter slot values to the feature structure produced by rules of the grammar, the grammar writer can explicitly express the expected measure of fragmentation that is associated with a particular grammar rule. For example, rules that combine fragments in less structured ways can be associated with higher counter values. As a result, analyses that are constructed using such rules will have higher counter values than those constructed with more structurally "grammatical" rules, reflecting the fact that they are more fragmented. In particular, the high level grammar rules that chain together SDU-level analyses can sum the fragmentation counter values of the individual SDU analyses that are being chained together.

5.2 Pruning Analyses Using Fragmentation Counters

The preference for a less fragmented analysis is realized by comparing the different analyses of SDU chains as they are being constructed, and pruning out all analyses that are not minimal in their fragmentation values. The pruning heuristic is implemented as a lisp procedure that is invoked along with the grammar rule that combines a new SDU analysis with a list of prior analyzed SDUs. The feature structure associated with the list of prior analyzed SDUs is pruned in a way that preserves only values that correspond to the minimum fragmentation. The feature structure of the new SDU is then combined only with these selected values.

Since the SDU combining grammar rule is invoked at each point where a part of the input utterance may be analyzed as a separate SDU, the pruning procedure incrementally restricts the parses being considered throughout the parsing of the input utterance. This results in a substantial decrease in the total number of ambiguous analyses produced by the parser for the given utterance, as well as a significant reduction in the amount of time and space used by the parser in the course of parsing the utterance.

5.3 Pruning Analyses Using Statistical Information

In addition to the fragmentation pruning, we use a statistical method that aims to prevent the parser from considering SDU boundaries at points in the utterance in which they are unlikely to appear. This is done using the same statistical information about the SDU boundary likelihood that is used for utterance pre-breaking. However, whereas in the pre-breaking process we attempt to detect locations in the utterance where an SDU boundary is likely to occur, within the parser we are attempting to predict the opposite, i.e., locations in which SDU boundaries are *unlikely*.

The likelihood of an SDU boundary is computed in the same fashion as previously described in Section 4. However, the information is now used differently. The procedure that calculates the SDU boundary likelihood is called by a special rule within the grammar, which is invoked whenever the parser completes a partial analysis that may correspond to a complete SDU. This grammar rule will only fire successfully if the point where the sentence ends is a statistically reasonable point to break the utterance. Should the rule fail, the parser will be prevented from pursuing a parse in which the next words in the utterance are interpreted as a new SDU. In order for the grammar rule to be allowed to fire, the computed boundary probability must be greater than a threshold set in advance. The value of the threshold is set manually so as to try and obtain as much pruning as possible, while not pruning out correct SDU segmentations. It is currently set to 0.03 for both Spanish and English.

To test the effectiveness of the statistical method of pruning out analyses, we compared the results of parsing an English test set of 100 utterances, both with and without statistical pruning. Using the statistical pruning resulted in an overall decrease of about 30% in parsing time. A comparison of the parse analyses selected by the parser showed that with statistical pruning, the parser selected a better parse for 15 utterances, while for 7 utterances a worse parse was selected. Although the seven bad cases are a result of missed SDU boundaries, the 15 good cases are a result of the parser selecting a better SDU segmentation, due to the fact that analyses with incorrect SDU boundaries were statistically pruned out.

5.4 Parse Disambiguation

All SDU segmentations allowed by the grammar that are not pruned out by the methods previously described are represented in the collection of analyses that are output by the parser. A parse disambiguation procedure is then responsible for the task of selecting the “best” analysis from among this set. Implicitly this includes selecting the “best” SDU segmentation of the utterance.

Disambiguation in GLR* 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. Each evaluation measure is a penalty function, which assigns a penalty score to each of the alternative analyses, according to its desirability. The penalty scores are then combined into a single score using a linear combination.

The parser currently combines three penalty scores. The first is a skip penalty that is a function of the words of the utterance that were not parsed in the course of creating the particular analysis. Different analyses may correspond to different skipped portions of the utterance. The penalty for skipping a word is a function of the word’s saliency in the scheduling domain. Highly salient words receive a high skip penalty. Analyses that skip fewer words, or words with lower saliencies are preferable, and thus receive lower penalties.

The fragmentation counter attached to the analysis is used as a second penalty score. As mentioned earlier, the value of the fragmentation counter slot reflects the amount of fragmentation of the analysis. For each of the parsable subsets of the utterance considered by the parser, pruning using fragmentation counters results in analyses that are minimal in the number of SDUs. In the disambiguation stage, where analyses of different parsable subsets are compared, the fragmentation counter is used as a penalty score, so as to once again reflect the preference for analyses that correspond to fewer SDUs.

The third penalty score is based on a statistical disambiguation module that is attached to the parser. The statistical framework is one 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 disambiguation. Statistical disambiguation can capture structural preferences in the training data. This will usually create a bias toward structures that correspond to SDU segmentations that are more likely to be correct.

6 Results

In order to test the effect of using pre-breaking on the output of the speech recognizer, we performed an end-to-end translation evaluation on a set of three unseen Spanish dialogues consisting of 103 utterances. These dialogues were never used for system development or training. The dialogues were first translated by both modules from unsegmented speech output and then from automatically segmented speech output. The results reported here are based on the percentage of acceptable translations of the 248 in-domain SDUs from the test set. These translations were scored by an objective independent scorer. Figure 5 shows a significant increase in translation accuracy for the GLR module from 36% to 54%. The Phoenix module is much less sensitive to the effects of pre-parsing segmentation. Thus, on unbroken utterances, Phoenix significantly out-performs GLR. However, pre-parsing segmentation results in a minor improvement in translation accuracy for the Phoenix translation module as well.

In Domain (248 SDUs)

	without pre-breaking	with pre-breaking
GLR	36%	54%
Phoenix	49%	52%

Figure 5. Results of a translation evaluation with and without pre-broken speech-recognition output.

7 Summary and Conclusions

Accurate speech translation in JANUS requires that an input utterance be correctly segmented into semantic dialogue units. We achieve this task using a combination of acoustic, statistical, lexical and semantic information, which is applied in two stages, prior to parsing and during parsing. Pre-parsing segmentation is advantageous because it increases the robustness of the parser to unparsable segments in the input utterance and significantly reduces the amount of segmentation ambiguity presented to the parser. However, accurate segmentation is performed during parse-time, when semantic and grammatical constraints can be applied. Pruning heuristics allow the parser to ignore segmentations that are unlikely to be correct. This restricts the set of possible analyses passed on for disambiguation. The disambiguation process subsequently selects the analysis deemed most correct.

REFERENCES

[1] R. Hausser. Principles of Computational Morphology. Technical Report, Laboratory for Computational Linguistics, Carnegie Mellon University, Pittsburgh, PA, 1989.

[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. Levin, D. Evans, and D. Gates. The ALICE System: A Workbench for Learning and Using Language. *Computer Assisted Language Instruction Consortium (CALICO) Journal*, Autumn 1991, 27–56.

[5] 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*.

[6] C. P. Rosé, B. Di Eugenio, L. S. Levin, and C. Van Ess-Dykema. Discourse processing of dialogues with multiple threads. In *Proceedings of ACL'95, Boston, MA*, 1995.

[7] M. Tomita and E. H. Nyberg 3rd. Generation Kit and Transformation Kit, Version 3.2: User's Manual. Technical Report CMU-CMT-88-MEMO, Carnegie Mellon University, Pittsburgh, PA, October 1988.

[8] M. Woszczyna, N. Aoki-Waibel, F. D. Buo, N. Coccaro, K. Horiguchi, T. 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.