Using Context in Machine Translation of Spoken Language

Lori Levin[†], Oren Glickman[†], Yan Qu[†], Donna Gates[†]
Alon Lavie[†], Carolyn P. Rosé[†], Carol Van Ess-Dykema[‡], Alex Waibel[†]
[†] Carnegie Mellon University (USA)
[‡] U.S. Department of Defense
lori.levin@nl.cs.cmu.edu

Abstract: We report on techniques for using discourse context to reduce ambiguity and improve translation accuracy in a multi-lingual (Spanish, German, and English) spoken language translation system. The techniques involve statistical models as well as knowledge-based models including discourse plan inference. This work is carried out in the context of the Janus project at Carnegie Mellon University and the University of Karlsruhe.

1 Introduction

Machine Translation of spoken language encounters all of the difficulties of written language (such as ambiguity) with the addition of problems that are specific to spoken language such as speech disfluencies, errors introduced during speech recognition, and the lack of clearly marked sentence boundaries. Fortunately, however, we can take advantage of the structure of task-oriented dialogs to help reduce these difficulties. In this paper we report on techniques for using discourse context to reduce ambiguity and improve translation accuracy in a multi-lingual (Spanish, German, and English) spoken language translation system. The techniques involve statistical models as well as knowledge-based models including discourse plan inference. This work is carried out in the context of the Janus project at Carnegie Mellon University and the University of Karlsruhe ([1]).

There has been much recent work on using context to constrain spoken language processing. Most of this work involves making predictions about possible sequences of utterances and using these predictions to limit the search space of the speech recognizer or some other component (See [2], [3], [4], [5], [6], [7], [8], [9]). The goal of such an approach is to increase the accuracy of the top best hypothesis of the speech recognizer, which is then passed on to the language processing components of the system. The underlying assumption being made is that design and complexity considerations require that each component of the system pass on a *single* hypothesis to the following stage, and that this can achieve sufficiently accurate translation results. However, this approach forces components to make disambiguation choices based solely on the level of knowledge available at that stage of processing. Thus, components of the system further down the line cannot correct a wrong choice of an earlier component.

s1: qué te parece el lunes	*how do you feel about Monday?*
s2: tal vez sería mejor en la tarde	*the afternoon* is *perhaps better*
corno a las a las dos de la tarde	*around two p.m.*
s1: no	*no*
yo tengo toda la tarde ocupada	*i am busy all afternoon*
de una a cuatro tengo una reunión	*from one o'clock till four o'clock i have a meeting*
s2: el lunes	*Monday*
entonces sería mejor el jueves	*then Thursday is better*

Figure 1: **Example of Translation**

The work reported in this paper does not rely on predictions about subsequent utterances (although we use such predictions in other work not reported here). The key feature of our approach is to allow multiple hypotheses to be processed through the system, and to use context to disambiguate between alternatives in the final stage of the process, where knowledge can be exploited to the fullest. Since it is unfeasible to process *all* hypotheses produced by each of the system components, context is also used locally to prune out unlikely alternatives. We describe four approaches to disambiguation, two of which are sentence-based and two of which are discourse-based in that, they take a multi-sentence context into account. We show that the use of discourse context improves performance on disambiguation tasks.

2 System Description

Janus is a speech-to-speech translation system currently dealing with dialogs in the scheduling domain (two people scheduling a meeting with each other). The current source languages are English, German, and Spanish and the target languages are English and German. We are also beginning to work with Korean, Japanese, and other languages. System development and testing is based on a collection of approximately 400 scheduling dialogs in each of the source languages. Translation of a portion of a transcribed dialog is shown in Figure 1.

The main modules of Janus are speech recognition, parsing, discourse processing, and generation. 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 2.[1]

Processing starts with speech input in the source language. Recognition of the speech signal is done with acoustic modeling methods, constrained by a language model. The output of speech recognition is a word lattice. We prefer working with word lattices rather than the more common approach of processing N-best lists of hypotheses. An N-best list may be largely redundant and can be efficiently represented in the form of a lattice. Using a lattice parser can thus reduce time and space

[1] Another approach being pursued in parallel in the Janus project is described in [10]

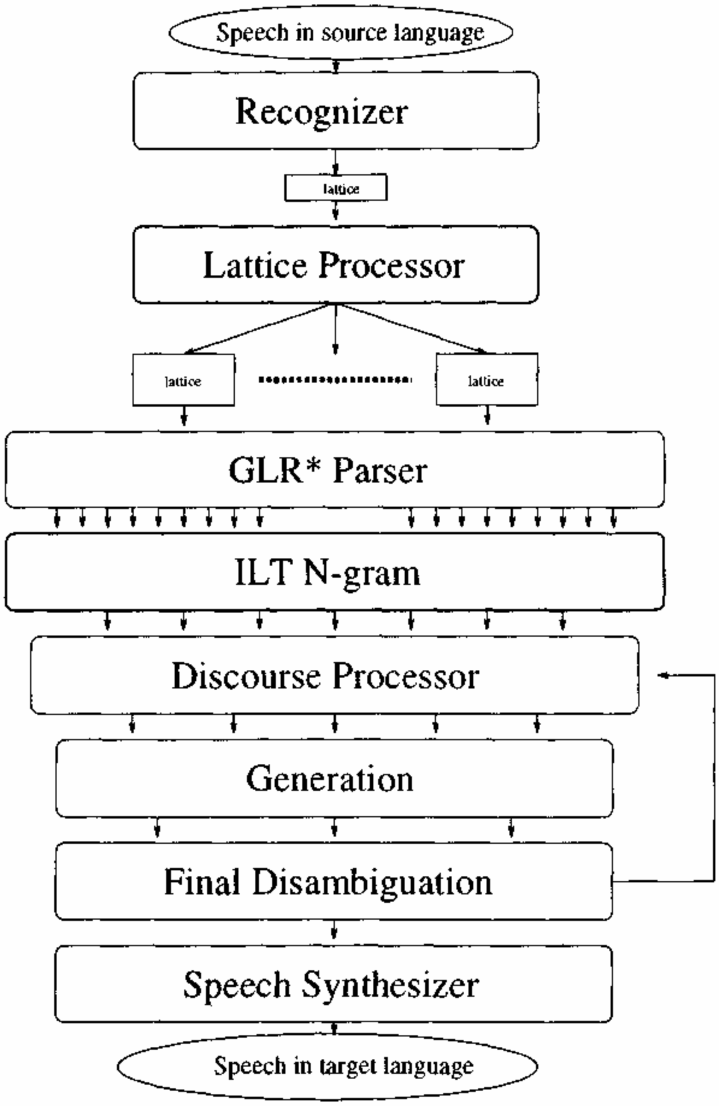


Figure 2: **Janus System Diagram**

complexity relative to parsing a corresponding N-best list. Selection of the correct path through the lattice is accomplished during parsing when more information is available.

Lattices, however, are potentially inefficient because of their size. We apply four steps to make them more tractable ([11]). The first step involves cleaning the lattice by mapping all non-human noises and pauses into a generic pause. Consecutive pauses are then adjoined to one long pause. The resulting lattice contains only linguistically meaningful information. The lattice is then broken at points where no human input is recognized over a specified threshold of time in the speech signal, yielding a set of sub-lattices which are highly correspondent to sentence breaks in the utterance. Each of the sub-lattices is then re-scored using a new language model. Finally the lattices are pruned to a size that the parser can process in reasonable time and space. The re-scoring raises the probability that the correct hypothesis will not be lost during the pruning stage. Each of the resulting sub-lattices are passed on to the parser, the first component of the translation process.

Parsing a word lattice involves finding all paths of connecting words within the lattice that are grammatical. The GLR* ([12], [13]) parser skips parts of the utterance that it cannot incorporate into a well-formed structure. Thus it is well-suited to domains in which extra-grammaticality is common. The parser can identify additional sentence breaks within each sub-lattice with the help of a statistical method that determines the probability of sentence breaks at each point in the utterance. The output of parsing a sub-lattice is a set of interlingua texts, or ILTs, representing all of the grammatical paths through the sub-lattice and all of the ambiguities in each grammatical path. The ILTs from each sub-lattice are combined, yielding a list of ILT sequences that represent the possible sentences in a full multi-sentence turn. An *ILT n-gram* is applied to each such list to determine the probability of each sequence of sentences.

The discourse processor, based on Lambert’s work ([14, 15]), disambiguates the speech act of each sentence, normalizes temporal expressions, and incorporates the sentence into a discourse plan tree. The discourse processor's focusing heuristics and plan operators eliminate some ambiguity by filtering out hypotheses that do not fit into the current discourse context. The discourse component also updates a calendar in the dynamic discourse memory to keep track of what the speakers have said about their schedules.

As processing continues, the N-best hypotheses for sequences of ILTs in a multi-sentence turn are sent to the generator. The generation output for each of the N hypotheses is assigned a probability as well. The generation output follows certain forms and is restricted in style. Therefore a regular n-gram model can be applied to assign a probability to each hypothesis.

The final disambiguation stage combines all knowledge sources obtained: the acoustic score, the parse score, the ILT n-gram score, information from the discourse processor, and a generation n-gram score. The best scoring hypothesis is sent to the speech synthesizer. This hypothesis is also sent back to the discourse processor so it can update its internal structures and the discourse state accordingly.

During translation, several knowledge structures are produced which constitute a

“Estás ocupada el lunes”
(Are you busy on Monday)

```
((FRAME *BUSY)
 (SENTENCE-TYPE *QUERY-IF)
 (A-SPEECH-ACT (*MULTIPLE* *SUGGEST
                           *REQUEST-RESPONSE))
 (SPEECH-ACT *REQUEST-RESPONSE)
 (WHO ((FRAME *YOU)))
 (WHEN
       ((WH    -)    (FRAME    *SIMPLE-TIME)
                    (SPECIFIER        DEFINITE)
                    (DAY-OF-WEEK MONDAY))))
```

Figure 3: **An Interlingua Text (ILT)**

discourse context that other processes can refer to. These knowledge structures include the ILT, the plan tree and focus stack, and the dynamically produced calendar. 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 3. The plan tree is based on a three-level model of discourse with discourse, domain, and problem solving levels. It shows how the sentences relate to each other in discourse segments. The focus stack indicates which nodes in the plan tree are available for further attachments. Figure 4 shows a plan tree at the discourse level. The first sentence, which is a surface question, is identified as a **Ref-Request** (request for information), a **Suggest-Form** (a possible way of making a suggestion), and finally part of an **Obtain-Agreement-Attempt** (a portion of the discourse in which the two speakers attempt to come to some agreement). The next sentence attaches as a **Self-Initiated-Clarification** indicating that this sentence makes the suggestion in the previous sentence more clear. The last two sentences are both **Accept-Forms** (acceptance of a suggestion) which chain up together to a **Response** node which then attaches to the corresponding suggestion. The Calendar records times which the speakers are considering, suggesting, rejecting, etc. This is updated dynamically as the conversation progresses. An example of a calendar is shown in Figure 5. Procedures that resolve ambiguity and select from among alternative analysis can take advantage of these knowledge structures as well as simpler ones such as the words in the previous sentence.

3 Techniques for Disambiguation

Resolution of ambiguity is important for accurate translation. Table 1 shows some examples of translation errors that are caused by failure to resolve ambiguity correctly. This section describes four disambiguation methods differing along two dimensions, whether they are knowledge-based or statistical, and whether they are sentence-based or take discourse context into account. The different types of ambiguities encountered in Spanish-to-English translation are summarized in Figure 6.

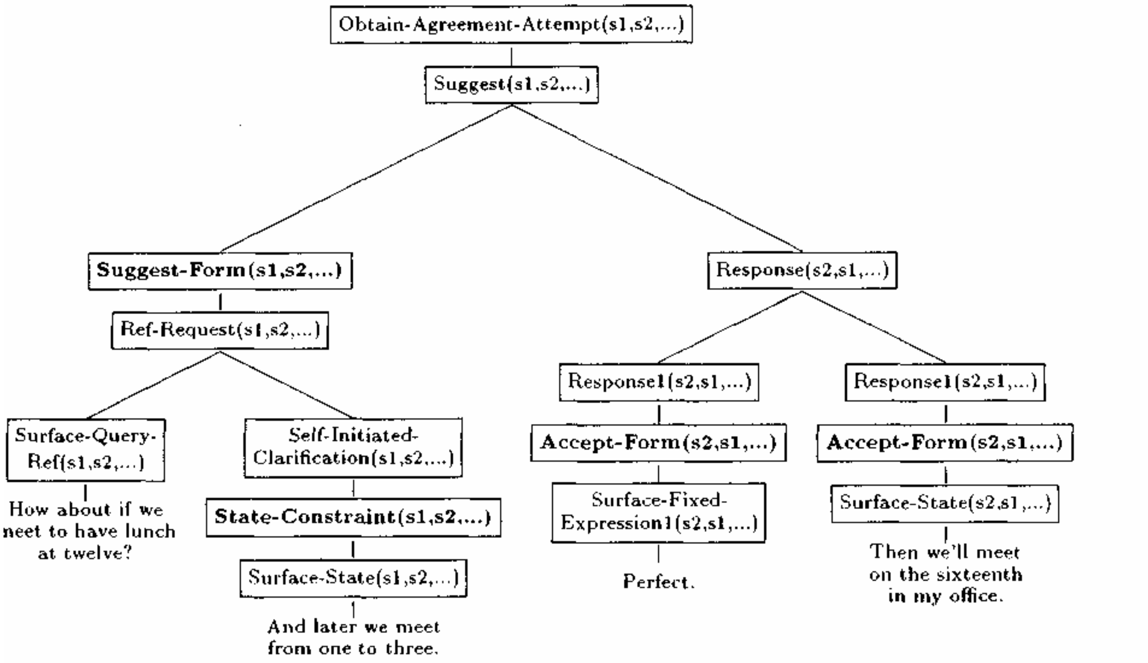
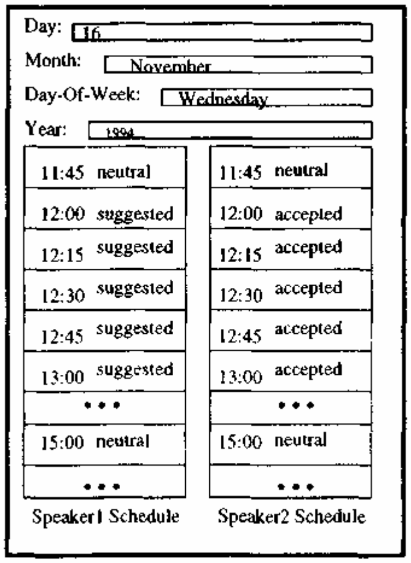

Obtain-Agreement-Attempt(s1,s2,...)
Suggest(s1,s2,...)
Suggest-Form(s1,s2,...)
Response(s2,s1,...)
Ref-Request(s1,s2,...)
Response1(s2,s1,...)
Response1(s2,s1,...)
Surface-Query-Ref(s1,s2,...)
Self-Initiated-Clarification(s1,s2,...)
Accept-Form(s2,s1,...)
Accept-Form(s2,s1,...)
How about if we neet to have lunch at twelve?
State-Constraint(s1,s2,...)
Surface-Fixed-Expression1(s2,s1,...)
Surface-State(s2,s1,...)
Surface-State(s1,s2,...)
Perfect.
Then we'll meet on the sixteenth in my office.
And later we meet from one to three.

Figure 4: **Example Plan Tree**



Day: 16

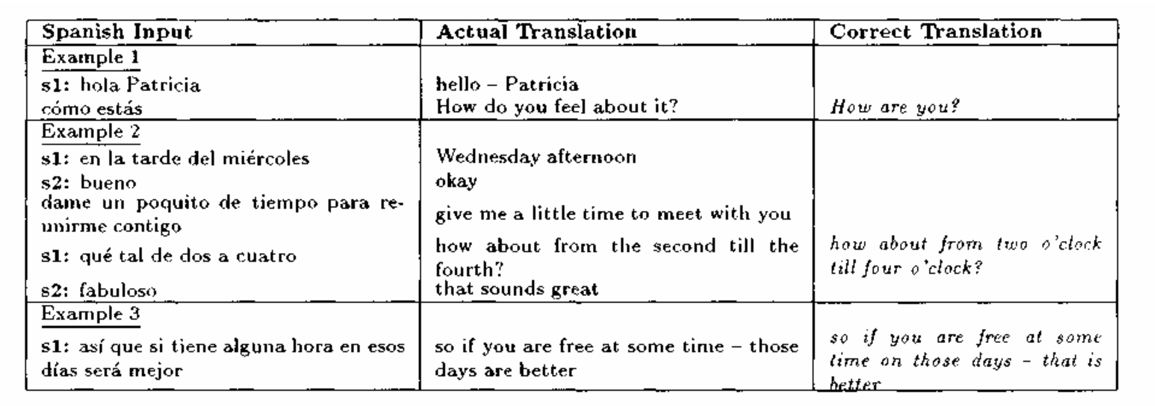
Month: November

Day-Of-Week: Wednesday

Year: 1994

Speaker1 Schedule	Speaker2 Schedule
11:45 neutral	11:45 neutral
12:00 *suggested*	12:00 accepted
12:15 suggested	12:15 accepted
12:30 suggested	12:30 accepted
12:45 suggested	12:45 accepted
13:00 suggested	13:00 accepted
• • •	• • •
15:00 neutral	15:00 neutral
• • •	• • •

Figure 5: **A Calendar Day Structure**



Spanish Input	Actual Translation	Correct Translation
Example 1		
s1: hola Patricia	hello – Patricia	
cómo estás	How do you feel about it?	*How are you?*
Example 2		
s1: en la tarde del miércoles	Wednesday afternoon	
s2: bueno	okay	
dame un poquito de tiempo para reunirme contigo	give me a little time to meet with you	
s1: qué tal de dos a cuatro	how about from the second till the fourth?	*how about from two o'clock till four o'clock?*
s2: fabuloso	that sounds great	
Example 3		
s1: así que si tiene alguna hora en esos días será mejor	so if you are free at some time – those days are better	*so if you are free at some time on those days – that is better*

Table 1: **Mistranslations of Ambiguous Sentences**

The following subsections describe the disambiguation methods that we tested. Our sentence-based disambiguation methods are implemented within the GLR* parser ([12] [13]) and its accompanying grammar. One method is knowledge-based, involving preferences that are explicitly encoded in grammar rules. The other is statistical, involving probabilities of actions in the LR parsing table. The context-based methods include knowledge-based discourse plan inference and statistical N-grams of ILTs.

Parse Disambiguation Using Grammar Rule Preferences

In order to successfully parse fragmented input, the grammars we use for parsing spontaneous speech have very inclusive notions as to what may constitute a "grammatical" sentence. The grammars allow meaningful clauses and fragments to propagate up to the top (sentence) level of the grammar, so that fragments may be considered complete sentences. Additional grammar rules allow an utterance to be analyzed as a collection of several grammatical fragments. The major negative consequence of this grammar "looseness" is a significant increase in the degree of ambiguity of the grammar. In particular, utterances that can be analyzed as a single grammatical sentence, can often also be analyzed in various ways as collections of clauses and fragments. Our experiments have indicated that, in most such cases, a less fragmented analysis is more desirable. Thus, we developed a mechanism for preferring less fragmented analysis.

The fragmentation of an analysis is reflected via grammar preferences that are set explicitly in various grammar rules. The preferences are recorded in a special *counter* slot in the constructed feature structure. 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. Although used to primarily reflect preferences with respect to fragmentation, the same mechanism can be used to express other preferences as well.

We tested the disambiguation performance of the GLR* parser using the grammar preferences as the sole disambiguation criterion. In this setting, for an ambiguous sentence that results in multiple analysis, the parser chooses the analysis with the lowest counter value. Ties between numerous analyses with equal minimal counter score are broken at random. This disambiguation method was tested on a set of 512 sentences, 252 of which produce ambiguous parses. As shown in Table 2, the GLR* parser using grammar rule preferences selected the correct parse in 196 out of the 252 ambiguous sentences. This corresponds to a success rate of 78%.

Parse Disambiguation Using a Statistical Model

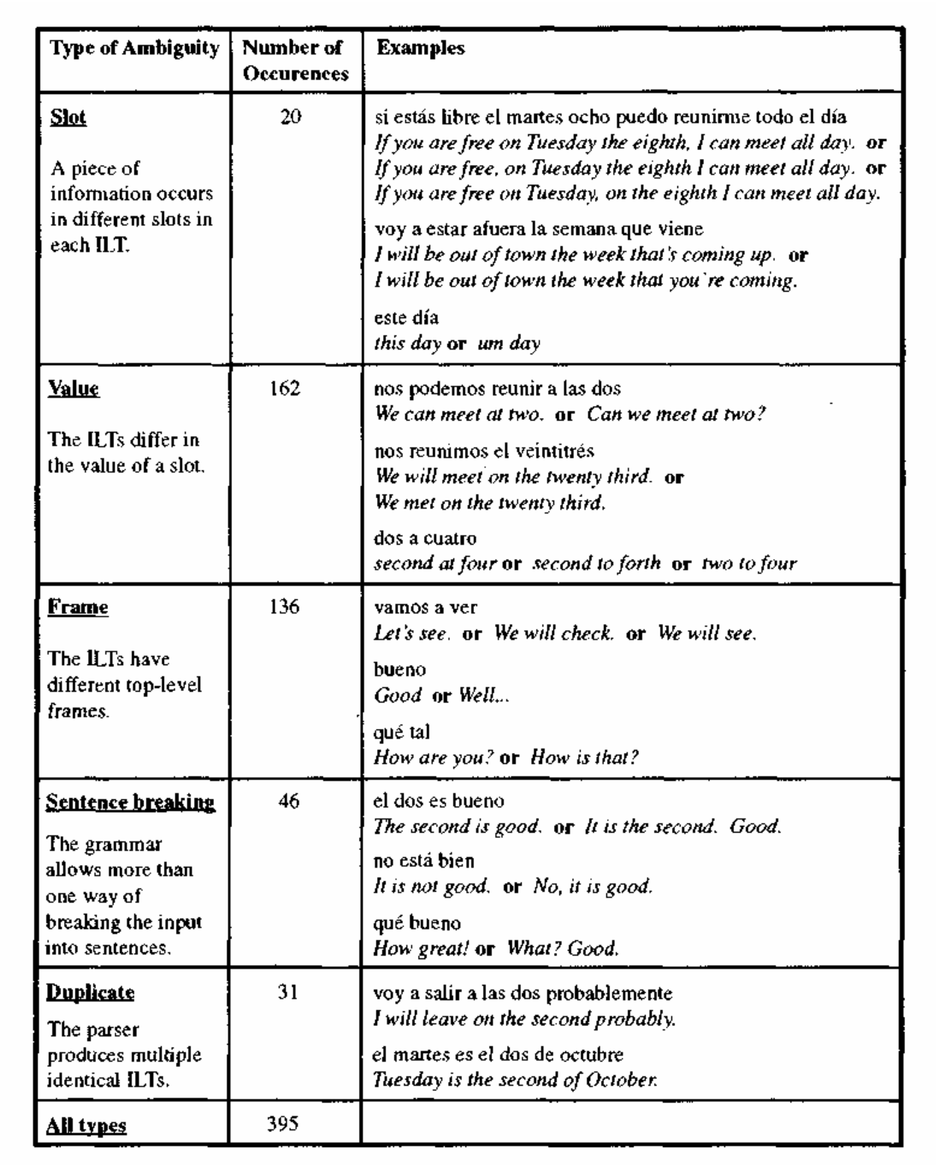
The grammar rule preference mechanism can reflect preferences between particular grammar rules. However, it does not provide a complete mechanism for disambiguating between the set of all possible analyses of a given input. This is done by a statistical module which augments the parser. Our statistical model attaches probabilities directly to the alternative actions of each state in the parsing table. Because the state of the GLR* parser partially reflects the left and right context within the sentence of the parse being constructed, modeling the probabilities at this level has the potential of capturing preferences that cannot be captured by standard Probabilistic Context-Free Grammars. For example, a reduce action by a certain grammar rule $A \rightarrow \alpha$ that appears in more than one state can be assigned a different probability in each of the occurrences.

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.

We tested the disambiguation performance of the GLR* parser using a combination of the statistical parse scores and the grammar rule preference values. The same test set of 252 ambiguous sentences was evaluated. As can be seen in Table 2, the combined disambiguation method succeeds in selecting the correct parse in 209 of the 252 cases, a success rate of 82%.

Disambiguation Using Discourse Plans

Our discourse processor is a plan inference model based on the recent work of Lambert ([14, 15]). The system takes as its input ILTs of sentences as they are uttered and relates them to the existing context, i.e., the plan tree. Plan inferencing starts from the surface forms of sentences. Then speech-acts are inferred. Multiple speech-acts for one ILT could be inferred. A separate inference chain is created for each possible speech act. Preferences for picking one inference chain over another are determined by the focusing heuristics, which provide ordered expectations of discourse actions



Type of Ambiguity	Number of Occurences	Examples
Slot A piece of information occurs in different slots in each ILT.	20	si estás libre el martes ocho puedo reunirme todo el día *If you are free on Tuesday the eighth, I can meet all day.* **or** *If you are free, on Tuesday the eighth I can meet all day.* **or** *If you are free on Tuesday, on the eighth I can meet all day.* voy a estar afuera la semana que viene *I will be out of town the week that's coming up.* **or** *I will be out of town the week that you're coming.* este día *this day* **or** *um day*
Value The ILTs differ in the value of a slot.	162	nos podemos reunir a las dos *We can meet at two.* **or** *Can we meet at two?* nos reunimos el veintitrés *We will meet on the twenty third.* **or** *We met on the twenty third.* dos a cuatro *second at four* **or** *second to forth* **or** *two to four*
Frame The ILTs have different top-level frames.	136	vamos a ver *Let's see.* **or** *We will check.* **or** *We will see.* bueno *Good* **or** *Well...* qué tal *How are you?* **or** *How is that?*
Sentence breaking The grammar allows more than one way of breaking the input into sentences.	46	el dos es bueno *The second is good.* **or** *It is the second. Good.* no está bien *It is not good.* **or** *No, it is good.* qué bueno *How great!* **or** *What? Good.*
Duplicate The parser produces multiple identical ILTs.	31	voy a salir a las dos probablemente *I will leave on the second probably.* el martes es el dos de octubre *Tuesday is the second of October.*
All types	395	

Figure 6: **Types of Ambiguities**

given the existing plan tree. A detailed description of the focusing heuristics can be found in [16] and [17].

We are currently conducting experiments to see how the plan tree and focusing heuristics can help to disambiguate multiple ILT outputs from the parser. We have obtained some preliminary results concerning resolving ambiguities in sentence types (**`statement`**, **`query-if`**, **`query-ref`**, **`fixed-expression`**, **`fragment`**) in the ILT outputs. Our experiments have shown that the same focusing heuristics, which are useful for picking the most preferred inference chain for one ILT, can be used for providing ordered expectations for picking inference chains from multiple ILT outputs of the parser.

The design of the experiment is composed of two steps. First, we try to attach each ILT from the set of ambiguous ILTs of a sentence to the existing dialog model. Second, the results of attachment for each ILT are compared. The best attachment is considered to be the one which best continues the existing context. When multiple attachments are possible, the focusing heuristics are used to make comparisons. For example, the sentence *Y nos podríamos reunir a la una* can be a statement *(And we could meet at one)* or yes-no question *(And could we meet at one?).* The focusing heuristic prefers the statement because it attaches to the current focus action, whereas the question attaches to an ancestor of the current focus action. The performance result of using plan tree and focusing strategy on sentence type ambiguities is shown in Table 3.

From Table 3, it can be seen that by using context and the focusing heuristics, the discourse processor achieves a general performance of 86% for sentence type disambiguation, which is an improvement over the 80% performance of the statistical parser without using context. For the **`statement`** vs **`query-if`** ambiguity, the discourse processor has a performance of 85%.

Statistical Methods for Using Context for Disambiguation

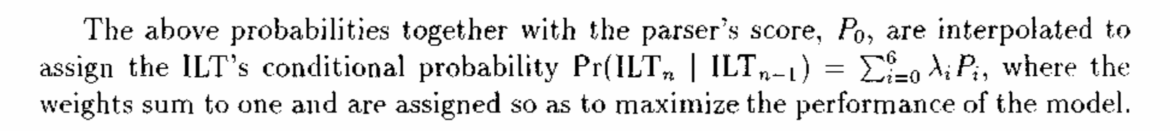
As we described above, the statistical scores assigned by the parser are based on sentence structure without taking the context of surrounding sentences into account. In this section we describe a statistical approach that uses context to help parse disambiguation. This work involved assigning probabilities to full utterances. We consider a full utterance, U, as a sequence of sentences represented by ILTs. Such an utterance could be assigned an approximated bigram probability by the formula:

$$\Pr(U) = \Pr(\mathrm{ILT}_1, \mathrm{ILT}_2, \ldots \mathrm{ILT}_n) = \prod_{i=1}^{n} \Pr(\mathrm{ILT}_i \mid \mathrm{ILT}_{i-1}) \qquad (1)$$

If ILT_i is the first ILT of an utterance, then ILT_{i-1} is the last ILT in the previous utterance of the other speaker.

Because we can not compute bigrams of full ILTs, our preliminary work has involved computing the probabilities of the **`sentence-type`**, **`speech-act`** and top-level **`frame`** of an ILT using the bigram probabilities described below. Standard smoothing techniques are used to calculate the conditional probabilities. Because we take into account the speakers of the current and previous sentences, a slot from the previous ILT is considered differently depending on if it was uttered by the same speaker or not. The amount of training data was not sufficient to calculate more complex N-grams such as Pr(frame$_n$ | frame $_{n-1}$ sentence-type$_{n-1}$ speech-act$_{n-1}$) or Pr(frame$_n$ | frame$_{n-1}$ frame$_{n-2}$) . We thus compute only the following probabilities:

P_1 = Pr(sentence-type$_n$ sentence-type$_{n-1}$)
P_2 — Pr(sentence-type$_n$ speech-act$_{n-1}$)
P_3 = Pr(sentence-type$_n$ | frame$_{n-1}$)
P_4 = Pr(frame$_n$ | sentence-type$_{n-1}$)
P_5 = Pr(frame$_n$ | speech-act$_{n-1}$)
P_6 = Pr(frame$_n$ | frame$_{n-1}$)



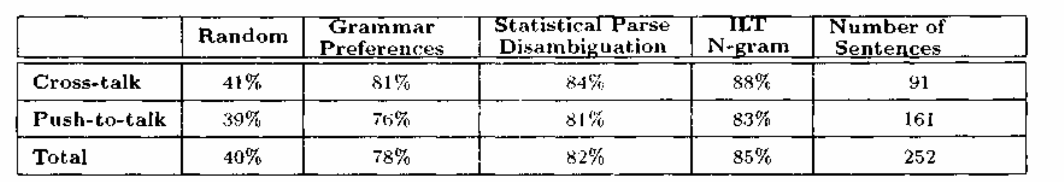
The above probabilities together with the parser's score, P_0, are interpolated to assign the ILT's conditional probability $\Pr(\mathrm{ILT}_n \mid \mathrm{ILT}_{n-1}) = \sum_{i=0}^{6} \lambda_i P_i$, where the weights sum to one and are assigned so as to maximize the performance of the model.

4 Comparison of Disambiguation Methods

Each of the disambiguation methods described above was trained or developed on a set of thirty Spanish scheduling dialogs and tested on a set of fifteen previously unseen dialogs. The development set and test set both contain a mixture of dialogs that were recorded in two different modes. In push-to-talk dialogs, participants cannot interrupt each other. The speaker must hit a key to indicate that he or she is finished speaking before the other participant can speak. In cross-talk dialogs, the participants can interrupt each other and speak simultaneously. Each speaker is recorded on a separate track. Push-to-talk sentences tend to be longer and more complex.

Table 2 shows the performance of three disambiguation methods in comparison to a baseline method of selecting a parse randomly. The three disambiguation methods are cumulative in the sense that each one builds on the previous one. The first method, Grammar Preferences, involves the explicit coding of preferences in grammar rules. The second method, Statistical Parse Disambiguation, refers to the parse score computed by the GLR* parser, which takes into account the probabilities of actions in the GLR* parsing table as well as the grammar preferences. The third method, ILT n-grams, disambiguates top-level frames, sentence-types, and speech-acts, but relies on the parse score to resolve other ambiguities. As can be seen in Table 2 and Figure 7, each method adds a slight improvement over the others that it incorporates.

Table 3 shows the performance of four disambiguation methods in resolving sentence-type ambiguities. The first row shows performance on the most common ambiguity in Spanish—the ambiguity between statements and yes-no questions (**`query-if`**). Without access to intonation, statements are often indistinguishable from yes-no questions because they have the same word order in some circumstances. The four methods compared are the Grammar Preferences, Statistical Parse Disambiguation, and



	Random	Grammar Preferences	Statistical Parse Disambiguation	ILT N-gram	Number of Sentences
Cross-talk	41%	81%	84%	88%	91
Push-to-talk	39%	76%	81%	83%	161
Total	40%	78%	82%	85%	252

Table 2: **Disambiguation of All Ambiguous Sentences**

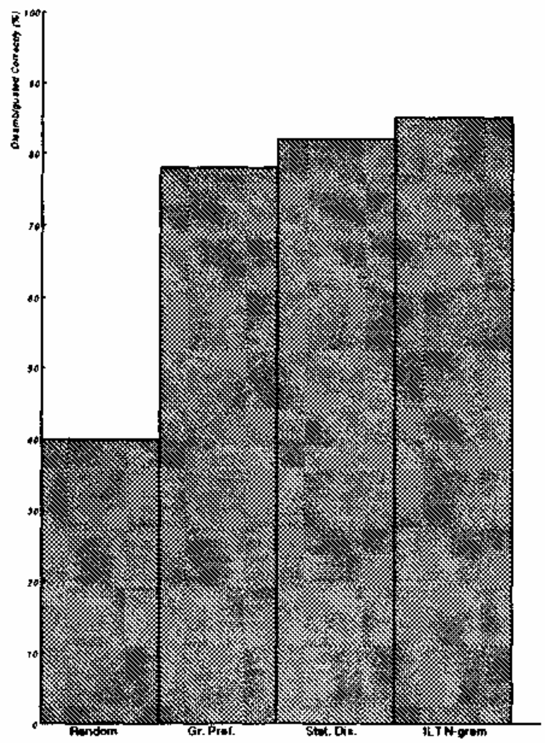
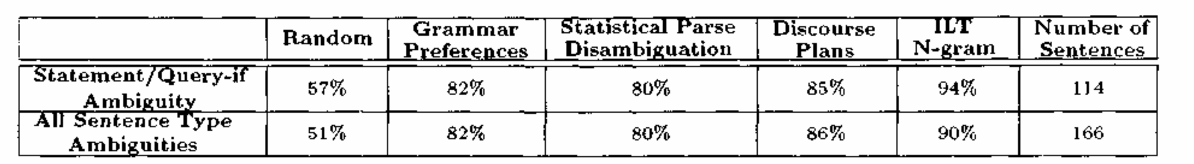


Figure 7: **Disambiguation of All Ambiguous Sentences**



	Random	Grammar Preferences	Statistical Parse Disambiguation	Discourse Plans	ILT N-gram	Number of Sentences
Statement/Query-if Ambiguity	57%	82%	80%	85%	94%	114
All Sentence Type Ambiguities	51%	82%	80%	86%	90%	166

Table 3: **Disambiguation of Sentence Types**

ILT N-grams described above, as well as Discourse Plan Inference. The Discourse Plan Inference is not cumulative with the other disambiguation methods. The input to the plan inference system is all of the ambiguous ILTs from the parser, without statistical parse scores. In this table, performance is calculated for the correct disambiguation of sentence-type only. Other ambiguities in the same sentences are not counted. The context-based methods, ILT N-grams and Discourse Plan Inference, perform better than the sentence-based methods in resolving the ambiguity between statements and yes-no questions. The second row of the table shows performance on all sentence-type ambiguities. Here also, the context-based methods do better than the sentence-based methods.

5 Conclusion

The approach we have taken is to allow multiple hypotheses and their corresponding ambiguities to cascade through the translation components, accumulating information that is relevant to disambiguation along the way. In contrast to other approaches that use predictions to filter out ambiguities early on, we delay ambiguity resolution as much as possible until the stage at which all knowledge sources can be exploited. A consequence of this approach is that much of our research effort is devoted to the development of an integrated set of disambiguation methods that make use of statistical and symbolic knowledge.

In this paper we examined four disambiguation methods, two that are sentence-based and two that use discourse context. In our experiments, the context-based methods performed somewhat better than the sentence-based methods. However, we believe that the best approach will be an integration of these and possibly other methods. Our future work will involve in particular how to combine the knowledge provided by the discourse processor with that provided by the parser and ILT N-grams. We believe that this is a promising path to follow because different sets of sentences are correctly disambiguated by each of the methods. Another feature of our future work will be to evaluate the effect of improved disambiguation on overall end-to-end translation quality.

References

[1] 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, In *ICASSP,* 1994.

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

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

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

[6] N. Reithinger, E. Maier. Utilizing Statistical Dialogue Act Processing in Verbmobil, In *Proceedings of the ACL,* 1995.

[7] N. Reithinger. Some Experiments in Speech Act Prediction, In *Proceedings of AAAI Spring Symposium,* 1995.

[8] J. Alexandersson, E. Maiser, N. Reithinger. A Robust and Efficient Three-Layered Dialogue Component for a Speech-to-Speech Translation System, In *Proceedings of the EACL,* Dublin, 1995.

[9] G. Hanrieder, G. Görz. Robust Parsing of Spoken Dialogue Using Contextual Knowledge and Recognition Probabilities, To appear in *Proceedings of ESCA Workshop on Spoken Dialogue Systems,* 1995.

[10] L.J. Mayfield. M. Gavalda, Y-H. Seo, B. Suhm, W. Ward, A. Waibel Concept-Based Parsing For Speech Translation, In Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation, 1995.

[11] Oren Glickman. Using Domain Knowledge to Improve End to End Performance in a Speech Translation System, Technical Report, Laboratory for Computational Linguistics, Carnegie Mellon University, 1995.

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

[13] A. Lavie. An Integrated Heuristic for Partial Parse Evaluation, In *Proceedings of 32nd Annual Meeting of the ACL,* 1994

[14] L. Lambert and S. Carberry. Modeling negotiation subdialogues, In *Proceedings of the ACL,* 1992.

[15] L. Lambert. *Recognizing Complex Discourse Acts: A Tripartite Plan-Based Models of Dialogue.* PhD thesis, University of Delaware, September 1993.

[16] C. P. Rosé. Plan-based discourse processor for negotiation dialogues. Unpublished ms, January 1995.

[17] C. P. Rosé, B. Di Eugenio, L. S. Levin, C. Van Ess-Dykema. *Discourse Processing of Dialogues with Multiple Threads* To appear in *Proceedings of the ACL,* 1995.