Using Discourse Predictions for Ambiguity Resolution

Yan Qu, Carolyn P. Rosé and Barbara Di Eugenio
Computational Linguistics Program
Department of Philosophy
Carnegie Mellon University
Pittsburgh, PA 15213
{yqu,cprose}@cs.cmu.edu,dieugeni@cmu.edu

Abstract

In this paper we discuss how we apply discourse predictions along with non context-based predictions to the problem of parse disambiguation in Enthusiast, a Spanish-to-English translation system (Woszcyna et al., 1993; Suhm et al., 1994; Levin et al., 1995). We discuss extensions to our plan-based discourse processor in order to make this possible. We evaluate those extensions and demonstrate the advantage of exploiting context-based predictions over a purely non context-based approach.

1 Introduction

A system which processes spoken language must address all of the ambiguities arising when processing written language, plus other ambiguities specific to the speech processing task. These include ambiguities derived from speech disfluencies, speech recognition errors, and the lack of clearly marked sentence boundaries. Because a large flexible grammar is necessary to handle these features of spoken language, as a side-effect the number of ambiguities increases. In this paper, we discuss how we apply discourse predictions along with non context-based predictions to the problem of parse disambiguation. This work has been carried out in the context of Enthusiast, a Spanish-to-English speech-to-speech translation system (Woszcyna et al., 1993; Suhm et al., 1994; Levin et al., 1995), which currently translates spontaneous dialogues between two people trying to schedule a meeting time.

A key feature of our approach is that it allows multiple hypotheses to be processed through the system in parallel, and uses context to disambiguate among alternatives in the final stage of the process, where knowledge can be exploited to the fullest extent. In our system, numerical predictions based on the more local utterance level are generated by the parser. The larger discourse context is processed and maintained by a plan-based discourse processor, which also produces context-based predictions for ambiguities. Our goal was to combine the predictions from the context-based discourse processing approach with those from the non context-based parser approach.

In developing our discourse processor for disambiguation we needed to address three major issues. First, most plan-based or finite state automaton based discourse processors (Allen and Schubert, 1991; Smith, Hipp, and Biermann, 1995; Lambert, 1993; Reithinger and Maier, 1995), including the one we initially developed (Rosé et al., 1995), only take one semantic representation as input at a time: thus, we had to extend the discourse processor so that it can handle multiple hypotheses as input. Secondly, we needed to quantify the disambiguating predictions made by the plan-based discourse processor in order to combine these predictions with the non context-based ones. Finally, we needed a method for combining context-based and non context-based predictions in such a way as to reflect not only which factors are important, but also to what extent they are important, and under what circumstances. We assume that knowledge from different sources provides different perspectives on the disambiguation task, each specializing in different types of ambiguities.

In this paper, we concentrate on the first two issues which are imperative to integrate a traditional plan-based discourse processor into the disambiguation module of a whole system. The third issue is very important for successful combination of predictions from different knowledge sources. We address this issue elsewhere in (Rosé and Qu, 1995).

The paper is organized as follows: First, we briefly introduce the Enthusiast speech translation system and discuss the ambiguity problem in Enthusiast. Then we discuss our discourse processor, focusing on those characteristics needed to generate predictions for disambiguation. Finally, we evaluate our performance, and demonstrate that the use of discourse context improves performance on disambiguation tasks over a purely non context-based approach in the absence of cumulative error.

2 System Description

The main modules of our system include speech recognition, parsing, discourse processing, and generation. Processing begins with the speech input in the source language. The top best hypothesis of the speaker's utterance is then passed to the parser. The GLR* parser (Lavie, 1995) produces a set of interlingua texts, or ILTs, for a given sentence. For robustness, the GLR* parser can skip words in the input sentence in order to find a partial parse for a sentence which otherwise would not be parsable. An ILT is a frame-based language independent meaning representation of a sentence. 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 1. The parser may produce many ILTs for a single sentence, sometimes as many as one hundred or more.

```
((when
     ((frame *simple-time)
      (day-of-week wednesday)
      (time-of-day morning)))
 (a-speech-act
     (*multiple* *suggest *accept))
 (who
     ((frame *i)))
 (frame *free)
 (sentence-type *state)))
```

Sentence: I could do it Wednesday morning too.

Figure 1: **An Example ILT**

The resulting set of ILTs is then sent to the discourse processor. The discourse processor, based on Lambert's work (Lambert and Carberry, 1992; Lambert, 1993), disambiguates the speech act of each sentence, normalizes temporal expressions from context, and incorporates the sentence into the discourse context represented by a plan tree. The discourse processor also updates a calendar which keeps track of what the speakers have said about their schedules. We will discuss the discourse processor and how we extended it for the disambiguation task in Section 4.

3 Ambiguity in Enthusiast

Because the spontaneous scheduling dialogues are unrestricted, ambiguity is a major problem in Enthusiast. We gauge ambiguities in terms of differences between members of the set of ILTs produced by the parser for the same source sentence. As we mentioned earlier, the disambiguation task benefits from both non context- and context-based methods. We observed that some classes of ambiguities can be more perspicuously dealt with in one way or the other.

3.1 Non Context-Based Disambiguation

When the parser produces more than one ILT for a single sentence, it scores these ambiguities according to three different non context-based disambiguation methods. The first method, based on (Carroll and Briscoe, 1993), assigns probabilities to actions in the GLR* parser's parse table. The probabilities of the parse actions induce statistical scores on alternative parse trees, which are then used for parse disambiguation. The resulting score is called the *statistical score*. The second method the parser uses to score the ILTs makes use of penalties manually assigned to different rules in the parsing grammar. The resulting score from this method is called the *grammar preference score*. The third score, called the *parser score*, is a heuristic combination of the previous two scores plus other information such as the number of words skipped. These three non context-based scores will be referred to later when we discuss combining non context-based predictions with context-based ones.

Error analysis of parser disambiguation output shows that the GLR* parser handles well ambiguities which are not strongly dependent upon the context for a reasonable interpretation. For example, the Spanish word *una* can mean either *one* or *a*, as an indefinite reference. The parser always chooses the indefinite reference meaning since the vast majority of training examples use this sense of the word. Moreover, since in this case incorrect disambiguation does not adversely affect translation quality, it makes sense to handle this ambiguity in a purely non context-based manner.

3.2 Context-Based Disambiguation

While a broad range of ambiguities can be handled well in a non context-based manner, some ambiguities must be treated in a context sensitive manner in order to be translated correctly. Table 1 lists some examples of these types of ambiguities. Each type of ambiguity is categorized by comparing either different slots in alternative ILTs or different values in ambiguous ILT slots given the same input utterance.

For example, one type of ambiguity best handled with a context-based approach is the *day* vs *hour* ambiguity, exemplified by the phrase *dos a cuatro*. It can mean either *the second at four*, *the second to the fourth* or *two to four*. Out of context, it is impossible to tell which is the best interpretation. Contextual information makes it possible to choose the correct interpretation. For example, if the speakers are trying to establish a date when they can meet, then *the second to the fourth* is the most likely interpretation. However,

Types of Ambiguity	Description	Examples
day vs hour	a temporal expression can be recognized as a day or an hour	dos a cuatro *second at four* **or** *second to fourth* **or** *two to four*
state vs **query-if**	ambiguity between sentence type **state** or **query-if**	está bien *It's OK* **or** *Is it OK?*
speaker reference	ambiguity between pro-drop pronouns	también podría ese día *also i could that day* **or** *also you could that day*
tense	ambiguity between past tense and present tense	dónde nos encontramos *where are we meeting* **or** *where were we meeting*
how vs **greet**	ambiguity between frame **how** and **greet**	qué tal *How are you?* **or** *How is that?*
when vs **where**	ambiguity between **when** slot and **where** slot	sábado quince *Saturday the fifteenth* **or** *Saturday building 15*

Table 1: **Examples of Context-Sensitive Ambiguities**

if the speakers have already chosen a date and are negotiating the exact time of the meeting, then only the meaning *two to four* makes sense.

Some sentence type ambiguities are also context-based. For example, *Está bien* can be either the statement *It is good* or the question *Is it good?*. This is an example of what we call the `state` vs `query-if` ambiguity: in Spanish, it is impossible to tell out of context, and without information about intonation, whether a sentence is a statement or a yes/no question. However, if the same speaker has just made a suggestion, then it is more likely that the speaker is requesting a response from the other speaker by posing a question. In contrast, if the previous speaker has just made a suggestion, then it is more likely that the current speaker is responding with an accepting statement than posing a question.

In general, we base our context-based predictions for disambiguation on turn-taking information, the stage of negotiation, and the speakers' calendar information. This information is encoded in a set of context-based scores produced by the discourse processor for each ILT.

4 Discourse Processing and Disambiguation

Context-based ranking of ambiguities is performed by the plan-based discourse processor described in (Rosé et al., 1995) which is based on (Lambert and Carberry, 1992; Lambert, 1993). Originally, our discourse processor took as its input the *single* best parse returned by the parser. The main task of the discourse processor was to relate that representation to the context, i.e., to the plan tree. In general, plan inference starts from the surface forms of sentences. Then speech-acts are inferred. Multiple speech-acts can be inferred for one ILT. A separate inference chain is created for each potential speech act performed by the associated ILT. Preferences for picking one inference chain over another were determined by the focusing heuristics, which provide ordered expectations of discourse actions given the existing plan tree. Our focusing heuristics, described in detail in (Rosé et al., 1995), are an extension of those described in (Lambert, 1993). In determining how the inference chain attaches to the plan tree, the speech-act is recognized, since each inference chain is associated with a single speech-act.

As mentioned in the introduction, for a plan-based discourse processor to deal with ambiguities, three issues need to be addressed:

1. The discourse processor must be able to deal with more than one semantic representation as input at a time. Note that simply extending the discourse processor to accept multiple ILTs is not the whole solution to the disambiguation problem: finer distinctions must be made in terms of coherence with the context in order to produce predictions detailed enough to distinguish between alternative ILTs.
2. Before context-based predictions can be combined with quantitative non context-based predictions, they must be quantified. It was necessary to add a mechanism to produce more detailed quantifiable predictions than those produced by the original focusing heuristics described in (Rosé et al., 1995).
3. Finally, context-based predictions must be combined successfully with non-context-based ones. The discourse processor must be able to weigh these various predictions in order to determine which ones to believe in specific circumstances.

Thus, we extended our original discourse processor as follows. It takes *multiple* ambiguous ILTs from the parser and computes three quantified discourse scores for each ambiguity. The discourse scores are derived by taking into account

attachment preferences to the discourse tree, as reflected by two kinds of focusing scores, and the score returned by the *graded constraints*, a new type of constraint we introduced. Then for each ambiguity the discourse processor combines these three kinds of context-based scores with the non context-based scores produced by other modules of the system to make the final choice, and returns the chosen ILT. As in the first version of the discourse processor, the chosen ILT is attached to the plan tree and a speech act is assigned to it. We discuss now how the discourse scores are derived. Note that lower values for all scores are preferred.

4.1 Focusing scores

The focusing scores are derived from focusing heuristics based on (Sidner, 1981; Lambert, 1993; Rosé et al., 1995). The focusing heuristics identify the most coherent relationship between a new inference chain and the discourse plan tree. Attachment preferences by the focusing heuristics are translated into numerical preference scores based on attachment positions and the length of the inference chains. The assignment of focusing scores reflects the assumption that the most coherent move in a dialogue is to continue the most salient focused actions, namely, the ones on the rightmost frontier of the plan tree. The first focusing score is a boolean *focusing flag*. It returns 0 if the inference chain for the associated ILT attaches to the rightmost frontier of the plan tree, 1 if it either attaches to the tree but not to the right frontier or doesn't attach to the tree. The second focusing score, the *focusing score* proper, assigns a score between 0 and 1 indicating how far up the rightmost frontier the inference chain attaches. The maximal score is assigned in the case that the inference chain does not attach.

4.2 Graded constraints

Once the discourse processor was extended to accept multiple ILTs as input, it became clear that for most ambiguous parses the original focusing heuristics did not provide enough information to distinguish among the alternatives. Our solution was to modify the discourse processor's constraint processing mechanism, making it possible to bring more domain knowledge to bear on the disambiguation task. In the original discourse processor, all of the constraints on plan operators, which we call *elimination constraints*, were used solely for the purpose of binding variables and eliminating certain inference possibilities. Their purpose was to eliminate provably wrong inferences, and in this way to give the focusing heuristics a higher likelihood of selecting the correct inference chain from the remaining set.

We introduced a different type of constraint, *graded constraints*, inspired by the concept of graded unification discussed in (Kim, 1994). Unlike elimination constraints, they neither bind variables nor eliminate any inferences. Graded constraints always return true, so they cannot eliminate inferences. However, they assign numerical penalties or preferences to inference chains based on domain specific information. This information is then used to rank the set of possible inferences left after the elimination constraints are processed.

For example, consider the day versus hour ambiguity we discussed earlier. In most cases inference chains for ILTs with this ambiguity have the same focusing scores. We introduce the `possible-time` constraint to check whether the temporal constraints conflict with the dynamic calendar or the recorded dialogue date when the inference chains are built. If the temporal information represented in an ILT is in conflict with the dialogue record date (e.g., scheduling a time before the record date) or with the temporal constraints already in the calendar (e.g., propose a time that is already rejected), a penalty score is assigned to that inference chain; otherwise, a default value (i.e. no penalty) is returned. Several graded constraints may be fired in one inference chain. Penalties or preferences for all graded constraints in the inference chain are summed together. The result is the graded constraint score for that ambiguity.

Introducing graded constraints has two advantages over adding more elimination constraints. As far as the system in general is concerned, graded constraints only give preferences, they do not rule out inferencing and attachment possibilities: thus, introducing new constraints will not damage the broad coverage of the system. As far as the discourse processor is concerned, it would be possible to achieve the same effect by adding more elimination constraints, but this would make it necessary to introduce more fine-tuned plan operators geared towards specific cases. By introducing graded constraints we avoid expanding the search space among the plan operators.

4.3 Combining Predictions

Once the information from the graded constraints and the focusing scores is available, the challenging problem of combining these context-based predictions with the non context-based ones arises. We experimented with two methods of automatically learning functions for combining our six scores into one composite score, namely a genetic programming approach and a neural net approach. The basic assumption of our disambiguation approach is that the context-based and non context-based scores provide different perspectives on the disambiguation task. They act together, each specializing in different types of cases, to constrain the final result. Thus, we want our learning approach to learn not only which factors are important, but also to what extent they are

important, and under what circumstances. The genetic programming and neural net approaches are ideal in this respect.

Genetic programming (Koza, 1992; Koza, 1994) is a method for "evolving" a program to accomplish a particular task, in this case a function for computing a composite score. This technique can learn functions which are efficient and humanly understandable and editable. Moreover, because this technique samples different parts of the search space in parallel, it avoids to some extent the problem of selecting locally optimal solutions which are not globally optimal.

Connectionist approaches have been widely used for spoken language processing and other areas of computational linguistics, e.g., (Wermpter, 1994; Miikkulainen, 1993) to name only a few. Connectionist approaches are able to learn the structure inherent in the input data, to make fine distinctions between input patterns in the presence of noise, and to integrate different information sources.

We refer the reader to (Rosé and Qu, 1995) for full details about the motivations underlying the choice of these two methods as well as the advantages and disadvantages of each.

5 Evaluation

Both combination methods, the genetic programming approach and the neural net approach, were trained on a set of 15 Spanish scheduling dialogues. They were both tested on a set of five previously unseen dialogues. Only sentences with multiple ILTs, at least one of which was correct, were used as training and testing data. Altogether 115 sentences were used for training and 76 for testing.

We evaluated the performance of our two methods by comparing them to two non context-based ones: a baseline method of selecting a parse randomly, and a Statistical Parse Disambiguation method. The Statistical Parse Disambiguation method makes use of the three non context-based scores described in Section 3. The two context-based approaches combine the three non context-based scores as well as the three context-based scores, namely the focusing flag, the focusing score, and the graded constraint score.

Table 2 reports the percentages of ambiguous sentences correctly disambiguated by each method. We present two types of performance statistics on the testing set: without cumulative error *Testing without CE* and with cumulative error *Testing with CE*. Cumulative error builds up when an incorrect hypothesis is chosen and incorporated into the discourse context, causing future predictions based on discourse context to be inaccurate. Notice that for the two non context-based approaches, the performance figures for both kinds of testing are the same because cumulative error is only an issue for context-based approaches.

Our results show that the discourse processor is indeed making useful predictions for disambiguation: when we abstract away the problem of cumulative error, we can achieve an improvement of 13% with the genetic programming approach and of 2.5% with the neural net approach over the parser's non-context based statistical disambiguation technique. For example, we were able to achieve almost perfect performance on the `state` vs `query-if` ambiguity, missing only one case with the genetic programming approach; thus, for this ambiguity, we can trust the discourse processor's prediction.

However, our results also indicate that we have not solved the whole problem of combining non context- and context-based predictions for disambiguation. In the face of cumulative error, both of the two discourse combination approaches suffer from performance degradation, though to a different extent. Our current direction is to seek a solution to the cumulative error problem. Some preliminary results in this regard are discussed in (Qu et al., 1996).

6 Conclusions

In this article we have discussed how we apply predictions from our plan-based discourse processor to the problem of disambiguation. Our evaluation demonstrates the advantage of incorporating context-based predictions into a purely non context-based approach. While our results indicate that we have not solved the whole problem of combining non context- and context-based predictions for disambiguation, they show that the discourse processor is making useful predictions and that we have combined this information successfully with the non context-based predictors.

Our current efforts are aimed at solving the cumulative error problem in using discourse context. We noticed that cumulative error is especially a problem in spontaneous speech systems where unexpected input, disfluencies, out-of-domain sentences and missing information cause the deterioration of the quality of context. One possibility is to reassess and reestablish the context state when a conflict is detected between context and other predictions. A second proposal is to keep the n-best hypotheses and to choose one only after having processed a sequence of inputs. Preliminary experiments show that both proposals help reduce the adverse effect of the cumulative error problem.

Our results also suggest another possible avenue of future development. Instead of trying to learn a general function for combining various information sources, we could decide which source of information to trust in a particular case and classify

	Training	Testing without CE	Testing with CE
Random	32%	45%	45%
Statistical Parse Disambiguation	76.5%	76.3%	76.3%
DP Genetic Programming	91.6%	89.5%	60%
DP Neural Net	85.2%	78.8%	71.3%

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

the type of ambiguity at hand with the best approach for this ambiguity. This could be accomplished, for example, with a decision tree learning approach.

Acknowledgements

The authors would like to thank Lori Levin, Alon Lavie and Alex Waibel for comments on the work reported here and thank the two anonymous reviewers for comments on the earlier version of the paper. The work is supported in part by a grant from the Department of Defense.

References

Allen, J. F. and L. K. Schubert. 1991. *The Trains Project.* Ph.D. thesis, University of Rochester, School of Computer Science.

Carroll, J. and T. Briscoe. 1993. Generalized probabilistic LR parsing of natural language (corpora) with unification-based grammars. *Computational Linguistics*, 19(1).

Sidner, C. L. 1981. Focusing for Interpretation of Pronouns. *American Journal of Computational Linguistics*, 7(4):217–231.

Kim, A. 1994. Graded unification: A framework for interactive processing. In *Proceedings of the Association for Computational Linguistics.*

Koza, J. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press.

Koza, J. 1994. *Genetic Programming II.* MIT Press.

Lambert, L. 1993. *Recognizing Complex Discourse Acts: A Tripartite Plan-Based Model of Dialogue.* Ph.D. thesis, Department of Computer Science, University of Delaware.

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

Lavie, A. 1995. *A Grammar Based Robust Parser For Spontaneous Speech.* Ph.D. thesis, School of Computer Science, Carnegie Mellon University.

Levin, L., O. Glickman, Y. Qu, D. Gates, A. Lavie, C. P. Rosé, C Van Ess-Dykema, and A. Waibel. 1995. Using context in machine translation of spoken language. In *Theoretical and Methodological Issues in Machine Translation.*

Miikkulainen, R. 1993. *Subsymbolic Natural Language Processing: An Integrated Model of Scripts, Lexicon, and Memory.* The MIT Press.

Qu, Y., B. Di Eugenio, A. Lavie, L. S. Levin, and C. P. Rosé. 1996. Minimizing Cumulative Error in Discourse Context. To appear in *ECAI Workshop Proceedings on Dialogue Processing in Spoken Language Systems.*

Reithinger, N. and E. Maier. 1995. Utilizing statistical dialogue act processing in Verbmobil. In *Proceedings of the ACL.*

Rosé, C. P., B. Di Eugenio, L. S. Levin, and C. Van Ess-Dykema. 1995. Discourse processing of dialogues with multiple threads. In *Proceedings of the ACL.*

Rosé, C. P. and Y. Qu. 1995. *Automatically Learning to Use Discourse Information For Disambiguation.* Center for Machine Translation, Carnegie Mellon University. Technical Report.

Smith, R. W., D. R. Hipp, and A. W. Biermann. 1995. An architecture for voice dialogue systems based on prolog-style theorem proving. *Computational Linguistics*, 21(3):218–320.

Suhm, B., L. Levin, N. Coccaro, J. Carbonell, K. Horiguchi, R. Isotani, A. Lavie, L. Mayfield, C. P. Rosé, C. Van-Ess Dykema, and A. Waibel. 1994. Speech-language integration in a multilingual speech translation system. In *Proceedings of the AAAI Workshop on Integration of Natural Language and Speech Processing.*

Wermpter, S. 1994. Connectionist learning of flat syntactic analysis for speech/language systems. In *Proceedings of the International Conference on Artificial Neural Networks.*

Woszcyna, M., N. Coccaro, A. Eisele, A. Lavie, A. McNair, T. Polzin, I. Rogina, C. P. Rosé, T. Sloboda, M. Tomita, J. Tsutsumi, N. Waibel, A. Waibel, and W. Ward. 1993. Recent advances in JANUS: a speech translation system. In *Proceedings of the ARPA Human Languages Technology Workshop.*