Augmented Role Filling Capabilities for Semantic Interpretation of Spoken Language

Lewis Norton, Marcia Linebarger, Deborah Dahl, and Nghi Nguyen

Unisys Center for Advanced Information Technology
Paoli, Pennsylvania 19301

ABSTRACT

This paper describes recent work on the Unisys ATIS Spoken Language System, and reports benchmark results on natural language, spoken language, and speech recognition. We describe enhancements to the system's semantic processing for handling non-transparent argument structure and enhancements to the system's pragmatic processing of material in answers displayed to the user. We found that the system's score on the natural language benchmark test decreased from 48% to 36% without these enhancements. We also report results for three spoken language systems, Unisys natural language coupled with MIT-Summit speech recognition, Unisys natural language coupled with MIT-Lincoln Labs speech recognition and Unisys natural language coupled with BBN speech recognition. Speech recognition results are reported on the results of the Unisys natural language selecting a candidate from the MIT-Summit N-best (N=16).

INTRODUCTION

Improving the performance of spoken language systems requires addressing issues along several fronts, including basic improvements in natural language processing and speech recognition as well as issues of integration of these components in spoken language systems. In this paper we report the results of our recent work in each of these areas. [1]

One major area of work has been in the the semantic and pragmatic components of the Unisys natural language processing system. The work in semantics enhances the robustness of semantic processing by allowing parses which do not directly express the argument structure expected by semantics to nevertheless be processed in a rule-governed way. In the area of pragmatics we have extended our techniques for bringing material displayed to the user into the dialog context to handle several additional classes of references to material in the display.

[1]This work was supported by DARPA contract N000014-89-C0171, administered by the Office of Naval Research. We are grateful to Victor Zue of MIT, Doug Paul of MIT Lincoln Laboratories and John Makhoul of BBN for making output from their speech recognition systems available to us. We also wish to thank Tim Finin, Rich Fritzson, Don McKay, Jim Meidinger, and Jon Pastor of Unisys and Lynette Hirschman of MIT for their contributions to this work.

In the area of integration of speech and natural language, we report on an experiment with three spoken language systems, coupling the same Unisys natural language system to three different speech recognizers as shown in Figure 1.

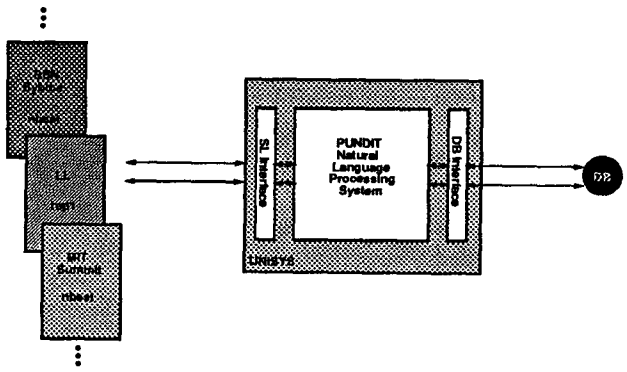

Figure 1: **Unisys natural language + multiple speech recognizers**

We believe this is a very promising technique for evaluating the components of spoken language systems. Using this technique we can make a very straightforward comparison of the performance of the recognizers in a spoken language context. Furthermore, this technique also allows us to make a fine-grained comparision of the interaction between speech and natural language in the three systems by looking at such questions as the relative proportion of speech recognizer outputs that fail to parse, fail to receive a semantic analysis and so on.

Finally, we report on speech recognition results obtained by filtering the N-best (N=16) from MIT-Summit through the Unisys natural language system. We note that there was a higher error rate for context-dependent speech as compared to context-independent speech (54.6% compared to 45.8%) and suggest two hypotheses which may account for this difference.

SEMANTICS

When evaluating our system after the Hidden Valley workshop, we observed two phenomena about PUNDIT (the Unisys natural language understanding system [3]) that warranted improvement. The first was that PUNDIT's semantic interpreter was sometimes failing to correctly recognize predicate

argument relationships for syntactic constituents that were not immediately associated with their intended head. The second was that PUNDIT was producing different representations for queries with different syntactic/lexical content but identical (or nearly identical) semantic content. We see both of these shortcomings as due to what we will term "non-transparent argument structure": syntactic representations in which syntactic constituents are not associated with their intended head, or semantic representations in which predicate-argument relationships are underspecified. Our approach to dealing with these shortcomings has been to maintain a rule-governed approach to role-filling despite non-transparent syntactic and semantic structures. We believe that the extensions we are about to describe are especially relevant to Spoken Language Understanding, because non-transparent argument structure appears to be particularly characteristic of spontaneous spoken utterances, for reasons we will sketch below.

The semantic interpreter and non-transparent parses

Semantic interpretation in PUNDIT is the process of instantiating the arguments of case frame structures called decompositions, which are associated with verbs and selected nouns and adjectives ([7]). The arguments of decompositions are assigned thematic role labels such as agent, patient, source, and so forth. Semantic interpretation is a search for suitable grammatical role/thematic role correspondences, using syntax-semantics mapping rules, which specify what syntactic constituents may fill a particular role; and semantic class constraints, which specify the semantic properties required of potential fillers. The syntactic constraints on potential role fillers are of two types: CATEGORIAL constraints, which require that the potential filler be of a certain grammatical type such as subject, object, or prepositional phrase; and ATTACHMENT constraints, which require that the potential filler occur within the phrase headed by the predicate of which it is to be an argument. The categorial constraints are stated explicitly in the syntax-semantics mapping rules; the latter are implicit in the functioning of the semantic interpreter. For example, the source role of flight_C, the domain model predicate associated with the noun "flight", can, in accordance with the syntax-semantics mapping rules, be filled by the entity associated with the object of a "from"-pp occurring within the same noun phrase as "flight" (*The flight from Boston takes three hours*). Unfortunately, the parse does not always express the argument structure of the sentence as transparently as it does in this example; constituents that should provide role fillers for a predicate are not always syntactically associated with the predicate. There are several causes for such a mismatch between the parse and the intended interpretation. They include (1) a variety of syntactic deformations which we will refer to as extraposition (*What flights do you have to Boston*, where the "to"-pp belongs in the subject np; *I need ticket information from Boston to Dallas*, where the pp's modify the prenominal noun "ticket", not the head noun "information"; or *I want a cheaper flight than Delta 66*, where the "than"-pp modifies "cheaper", not "flight"), (2) metonymy (*I want the $50.00 flight*, where the speaker means that s/he wants the flight whose FARE is $50.00), and (3) suboptimal parses (e.g., parses with incorrect pp-attachment).

Our changes to the semantic interpreter allow it to fill roles correctly in cases such as the above, utilising its existing knowledge of syntax-semantics correspondences, but relaxing certain expectations about the syntactic attachment of role-filling constituents. Thus the CATEGORIAL constraints remain in force, but the ATTACHMENT constraints have been loosened somewhat. The system now identifies prepositional phrases and adverbs which have not filled a role in the predicate with which they are syntactically associated, and offers them as role fillers to fillers of this predicate. This strategy applies recursively to fillers of fillers of roles; for example, in *What types of ground transportation services are available from the airport in Atlanta to downtown Atlanta?*, the two final pp's ultimately fill roles in the decomposition associated with "ground transportation" since neither "types" nor "services" has mapping rules to consume them. The same mechanism already in place for role-filling is employed in these cases, the only difference being that unused syntactic constituents are propagated downward. Note that we continue to take syntax into account; we do not wish to ignore the syntax of leftover constituents and fill roles indiscriminately on the basis of semantic properties alone.

We conducted an experiment to assess the effects of these changes upon the system's performance, using a set of 138 queries (both class A and non-class A) on which the system was previously trained. The measure of performance used was the standard ATIS metric of the number of correct answers minus the number incorrect. Disabling the semantic changes described above lowered the system's score from 82 to 63, a decrease of 23%.

The application module and non-transparent semantic representations

Our second improvement was directed at cases where PUNDIT's semantic interpreter may have correctly represented the meaning of a sentence but in an irregular way. For example, the instantiated decomposition produced for "flights from Boston" is:

```
flight_C(flight1, source(boston), ...)
```

while "flights leaving Boston" resulted in:

```
flight_C(flight1, source(_), ...)
leaveP(leave1,
       flight(flight1),
       source(boston), ...)
```

Clearly it would be preferable for the flight_C decomposition to be the same in both cases, but in the second case the source role of the decomposition associated with flight1 was unfilled, although it could be inferred from the leaveP decomposition that the flight's source was Boston. In other words, PUNDIT had not captured a basic synonymy relation between these np's.

Our response to this was to augment the semantic interpreter with a routine which can perform inferences involving more than one decomposition. The actual inferences are expressed in the form of rules which are domain-dependent; the inference-performing mechanism is domain-independent. For the above example, we have written a rule which, paraphrased

in English, says that if a verb is one of a class of motion verbs used to express flying (e.g., "leave"), and if the source role of this verb is filled, propagate that filler to the source role of the flight involved. Thus the flight_C decomposition becomes the same for both inputs. Thirty-four such rules have been written for the ATIS domain, and we estimate that they are applicable to 10% to 15% of the training data.

The payoff from this extension comes in the use of PUNDIT's output by application modules. For the ATIS domain, the application module is the program that takes PUNDIT's output and uses it to formulate a query to the ATIS DB. It is obviously advantageous for the creation and maintenance of an application module that its input be regularized to the greatest extent possible, thus making such a module simpler, and avoiding duplication of code to compensate for non-regularized input in different application modules.

When we ran the same set of 138 queries used in the experiment described in the previous subsection without the rules just discussed (but with the semantics improvements of the previous subsection), the system's score dropped from 82 to 62, or 24%. There appears to be little interaction between the semantics improvements and the rules of this subsection–they apply to different phenomena in input data.

PRAGMATICS

In our June 1990 workshop paper ([6]), we described a feature of our system which we included to handle correctly a particular kind of discourse phenomenon. In particular, in the ATIS domain there are frequent references to flights by flight number (e.g., "Delta flight 123") which the user means to be unambiguous, but which in general have to be disambiguated in context. The reason is that the user learned about "Delta 123" from some previous answer, where it was returned as one of the flights between two cities City1 and City2. The problem is that "Delta 123" may have additional legs; for instance it may go on from City2 to City3. The user, when asking for the fares for "Delta 123", is presumably interested only in the City1 to City2 fare, not the City2 to City3 one and not the City1 to City3 one. So our system looked back at previous answers to find a mention of "Delta 123", thereby determining the flight leg of interest.

This kind of disambiguation can take other forms, and we have added some of them to our system since June. One of these capabilities is illustrated by the two queries *What does LH mean?* and *What does EQP mean?* Without context, the first of these cannot be correctly answered, because "LH" is a code for both an airline and a fare class. The second of these queries would yield a table with two rows, one row for each table for which "EQP" is one of the table's column headings. In both of these queries, however, the user is asking for clarification of something which has been presented as part of a previous answer display. So what our system needs to do, and does do, is refer back to previous answers much in the spirit of the "Delta 123" example above. For the first query, we will find the most recent answer which has "LH" as a column entry in some row: for the second we will find the most recent answer which has "EQP" as a column heading. Our system can then make the proper disambiguation and present the user with an appropriate cooperative response to the follow-up query. There were only a handful of follow-up queries of this form in the training data, but the extension to handle them was easy to add given the code in place to handle the "Delta 123" example.

Similarly, the training data contained numerous instances of queries such as *What are the classes?* In the absence of context, the best answer to this seems to be a list of the more than 100 different fare classes. However, queries such as these invariably follow the display of some fare classes in either flight tables or fare tables. The cooperative response, then, is to display a table of fare classes whose rows have been limited to those classes previously mentioned in the most recent flight or fare table. Our system also uses a generalization of this algorithm to filter requests for other kinds of codes, such as restrictions, ground transportation codes, aircraft codes, and meal codes. In all, from the TI training data ([2]) we have noticed 19 follow-up queries (out of 912) which now get the correct answer in context because of this extension to our system; there may be more queries which require this extension that we have not yet processed correctly for other reasons.

We make it possible to refer to previous answer tables in our system by means of the following mechanism. Whenever an answer table is returned, a discourse entity representing it is added to the discourse context, and a semantics for this entity is provided. Roughly speaking, if the query leading to the answer table is a request for X, the semantics can be thought of as being "the table of X" ([6]). For example, if the query was a request for flights from City1 to City2, the semantics assigned to the discourse entity representing the answer is "the table of flights from City1 to City2". Note that we do NOT create discourse entities for each row (particular flights from City1 to City2 in the example) or for each column entry in a row (e.g., the departure time of a particular flight from City1 to City2). Doing so would make the discourse context unmanageably large. But the table (complete with column headings) is available and accessible to our system, and can be searched for particular values when it is desirable to do so, as in the capabilities being described in this section.

The techniques just described depend on the availability of previous ANSWERS. Some of the follow-up queries which they enable to be answered correctly could perhaps be handled by reference to previous QUERIES only, particularly in the special case where there is known to be only one previous query. We believe that our techniques are superior for at least two reasons. First, in the presence of more than one previous query, the answers to those previous queries are for our system a more compact and modular representation of the content of those queries than the discourse entities created while analyzing the queries themselves; in short, it is simply easier to find what we want in the answers rather than in our representations of the queries. Second, there are follow-up queries which cannot be answered unless reference is made to previous answers, so such techniques are necessary in a complete system. Therefore, why not use them whenever they can be used, even when alternative techniques might be available?

The February 1991 D1 pairs test, which limited context dependency to dependency which could be resolved by examination of a single previous query (and not its answer), provides

additional data on the applicability of these methods. In particular, 27 of the 38 pairs involved the disambiguation of a flight number to the flight leg of interest. It appears that four additional queries can be successfully answered by the technique we discussed above for handling the query *What are the classes?* The remaining 7 queries appear to be such that reference to previous answers is not helpful.

SPOKEN LANGUAGE SYSTEMS

We describe here the five spoken language tests in which we participated. Our methodology in these tests has been to couple the speech recognition output from different recognizers to the same natural language processing system. Because the natural language component and the application module are held constant in these systems, this methodology provides us with a means of comparing the performance of speech recognizers in a spoken language context.

Class A: Unisys PUNDIT system coupled with MIT Summit

The spoken language system used in this test consists of the Unisys PUNDIT natural language system coupled via an N-best interface to the MIT SUMMIT speech recognition system. We will refer to this system as Unisys-MIT. These results were run with N=16, except for 4 utterances which could not be run at N=16 because of insufficient memory in the speech recognition system. N=1 was used for these utterances. SUMMIT produced the N-best and PUNDIT selected from the N-best the most acoustically probable utterance which also passed PUNDIT's syntactic, semantic, and pragmatic constraints. PUNDIT then processed the selected candidate to produce the spoken language system output. The value of N of 16 was selected on the basis of experiments reported in [1], which demonstrated that using larger N's than 10-15 leads to a situation where the chance of getting an F begins to outweigh the possible benefit of additional T's.

The SUMMIT system is a speaker-independent continuous speech recognition system developed at the MIT Laboratory of Computer Science. It is described in [10].

Unisys PUNDIT coupled with Lincoln Labs Speech Recognizer

The spoken language system used in this test consists of the Unisys PUNDIT natural language system loosely coupled to the MIT Lincoln Labs speech recognition system. The Lincoln Labs system selected the top-1 output, which PUNDIT then processed to produce the spoken language system output.

The Lincoln Labs system is a speaker independent continuous speech recognition system which was trained on a corpus of 5020 training sentences from 37 speakers. It used a bigram backoff language model of perplexity 17.8. The system is described in more detail in [8].

Class A: Unisys PUNDIT system coupled with BBN BYBLOS

In this test N-best output from the BBN BYBLOS system as described in [4] was input to PUNDIT. As in the system which used the MIT N-best, we used an N of 16. The N-best from BBN was the output from BYBLOS rescored using cross-word models and a 4-gram model and then reordered before input to the natural language system.

Class	*Number*	T	F	*Score*
Class A	145 queries	84	14	48.3%
Class D1	38 pairs	24	0	63.2%
Class AO	11 queries	1	0	9.1%
Class D1O	2 pairs	0	1	-50%

Table 1: **Unisys System Scores**

Optional Class A Tests

We also report on spoken language results on the optional class A test, using both the Unisys-MIT system and the Unisys-BBN systems described above.

SPEECH RECOGNITION TESTS

The speech recognition tests were done using the natural language constraints provided by the Unisys PUNDIT natural language system to select one candidate from the N-best output of the MIT Laboratory of Computer Science SUMMIT speech recognition system. Using an N of 16, PUNDIT selected the first candidate of the N-best which passed its natural language constraints based on syntactic, semantic and pragmatic knowledge. If all candidates were rejected by the natural language system, the first candidate in the N-best was considered to be the recognised string.

BENCHMARK RESULTS

Natural Language Common Task Evaluation

Unisys attempted all four of the natural language tests; both the required and the optional class A and class D1 tests. Our scores as released by NIST are as shown in table 1. The overall level of success is unimpressive. For the class A test, which corresponds most closely to the test last June, our performance is not much better, in spite of eight more months of work on our system. (If the scoring algorithm in effect now had been in effect in June, our score then would have been 42.2%) As this paper is being written, we have not had the time to examine our performance on a sentence by sentence basis. It appears likely, however, that the amount of training data has not yet adequately covered the full range of the various ways that people can formulate queries to the ATIS database.

We are fairly pleased that our "false alarm" rate has not gone up since June. It was 11% then; if we take the 196 sentences involved in the latest 4 tests as a single group, we find our rate of F's to be less than 8%. When we discuss our spoken language results in a subsequent section, we will see that although the rate of correct answers drops noticeably when a speech recognizer is added to the system, the rate of incorrect answers does not appear to increase. The importance of a low

"false alarm" rate is well appreciated by spoken language understanding researchers; from a user's point of view nothing could be worse than an answer which is wrong although the user may have no way of telling it is wrong. It will be important to lower the rate of such errors to a level well below 8%.

Our best performance came on the D1 pairs test. One would have expected a lower score on any test that requires two consecutive sentences to be understood than on a test of self-contained sentences. While we wish we could claim that our work discussed in the earlier section on pragmatics was instrumental in achieving our score, it appears that much of what we added to our system did not come into play in this test. A more likely explanation of the unexpectedly high score is that when a user queries the system in a mode which utilizes follow-up queries, he or she tends to use simpler individual queries. Perhaps a user who does not use follow-up queries is trying to put more into each individual query. Some evidence for this is that our score for just the 20 distinct class A antecedent sentences for the D1 pairs test was 75%, well above our 48.3% score for all the class A sentences. Even more striking is the fact that of the 9 speakers represented in this round of tests, only two contributed more than 3 pairs to the class D1 test–speakers CK and CO contributed 13 pairs each. Our scores restricted to just those two speakers were 93% for the class A test and 65% for the class D1 test (100% for speaker CO in the class D1 test!).

The optional tests clearly were too small to have much significance. It is not surprising that our system proved to be incapable at this point of dealing with extraneous words in the input queries, for we have made no efforts as yet to compensate for such inputs. These tests will be useful as a benchmark for comparison after we have addressed such issues.

Semantics Extensions and the Common Task Tests

In the section on semantics we reported the results of two experiments that we ran to assess the effects of extensions to our system. We performed the same tests using the data of the latest class A test of 145 queries. When the extensions to our semantic interpreter were removed, our performance dropped to 72 T, 19 F, or 36.6%, a decrease of 24% from our score of 48.3%. This reinforces our belief that these extensions are very important and useful. When we ran the test without the rules relating multiple decompositions, our performance was 83 T, 14 F, or 47.6%, a decrease of less than 2%. This latter finding was most surprising–basically it implies that in the 1991 test data there were virtually no constructions of the kind which those rules enable us to process, because the absence of the rules relating the decompositions corresponding to those constructions resulted in almost no reduction in our score. In particular, there must have been no nouns modified by relative clauses ("flights that arrive before noon") or participial modifiers ("flights serving dinner"). This has some implication regarding the distribution of various forms of syntactic expression across speakers, for phenomena which were clearly significant in our training data apparently were absent from 9 speakers' worth of test data.

The above experiments imply that our system as of last June would have gotten a score of less than 35% on the current class A test, for the extensions discussed in the section on semantics were not the only improvements we have made to our system. This is another indication of variability among speakers; for our system the 5 speakers of last June's test were easier to process. It appears to us that larger test sets are necessary to make a broad evaluation of natural language understanding capabilities. (We do not extend this suggestion to tests involving speech input because of the level of effort that would consume.) We have already noted the absence of relative clauses and participial modifiers in the recent class A test. We also noticed that 23 of 145, or 16%, of the sentences used the word "available", usually in constructions like "what X is available", while this word only appeared in 4% of the pilot training data. In the class D1 test, there were few discourse phenomena represented, and we noted in an earlier section that over 70% (27 of 38) of the D1 pairs involved just the phenomenon of flight leg disambiguation. Tests of such size, then, are not broadly representative of the range of query formulations in the ATIS domain.

Related to the last point is the suspicion that the few thousand sentences of training data are themselves too few to represent the range of user queries for this domain. We have noticed that fewer new words are appearing in the more recent sets of training data, so vocabulary closure is probably occurring. Even so, in the 145 class A queries of the recent test, our system found 12 with unknown words, or 8% of the queries. This was actually higher than the 5.5% we experienced with the test last June, but that is more a comment on the variability due to small test size. It is an open question whether more and more training data is the answer to making our systems more complete, however. After all, larger volumes of data are both expensive to collect and expensive to train from. The lack of closure for the syntactic and semantic variation in user queries presents a challenge for further research in spoken language understanding. It may well be that we will have to begin studying reasonable ways in which the variation in the range of user expression can be limited, without unduly contraining the user in the natural performance of the task.

Spoken Language Evaluations

Unisys-MIT The spoken language results for this system were 29 T, 15 F, and 101 NA, for a weighted score of 9.7%. The system examined an average of 6.5 candidates in the N-best before finding an acceptable one. Of all candidates considered by the system, we found that 85% were rejected by the syntax component and 3% by the semantics/pragmatics component, and 11% were accepted by both components. It should be pointed out that the syntax component uses a form of compiled semantics constraints during its search for parses ([5]), thus the results for purely syntactic rejection are not as high as appears in this comparison, because some semantic constraints are applied during parsing. After a candidate is accepted by both syntax and semantics, the search in the N-best is terminated. However, the application component, which contains a great deal of information about domain-specific pragmatics, can also reject syntactically and semantically acceptable inputs for which it cannot construct a sensible database query. In fact, a syntactically and semantically acceptable candidate was found in 75% of the N-best candidate lists, but a call was

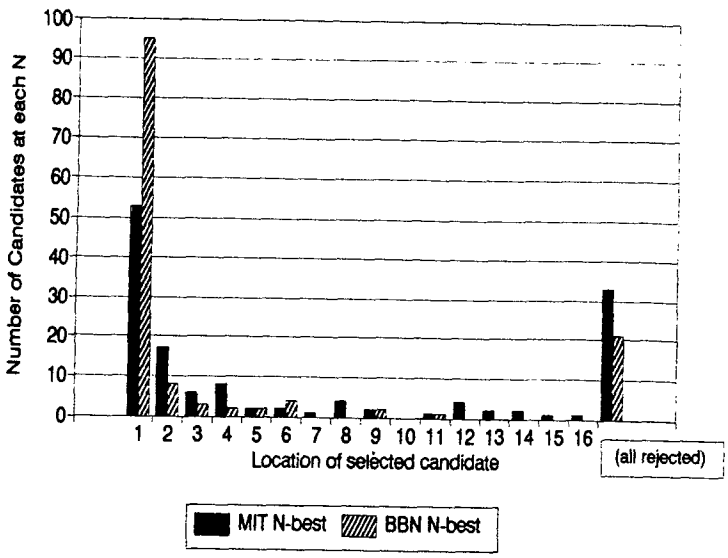

Figure 2: Comparison of Location of Accepted Candidate in N-best (N=16) for Unisys-MIT and Unisys-BBN Systems

made for only 30% of inputs. The application component was not able to make a sensible call for the remaining inputs.

The false alarm (or F) rate we observed in this test was around 10%, which is consistent with our previous spoken language results ([1]) and with our natural language results, as discussed above.

Unisys-BBN This system received a score of 77 T, 20 F and 48 NA for a weighted score of 39.3%. In this system 74% of all inputs were rejected by syntax, 11% of inputs were accepted by syntax but rejected by semantics and 15% were accepted by both syntax and semantics. The false alarm rate is 13.8%, which is slightly higher but in the same range as previous false alarm rates.

As can be seen in Figure 2, in general the system found an acceptable candidate earlier in the N-best with the BBN N-best than with the MIT N-best. The average location of the selected candidate in the N-best with the BBN data was 3.8 compared to 6.5 with the MIT N-best.

Unisys-LL Using the top-one candidate from the Lincoln Labs speech recognizer the spoken language results for this system were 32 T, 5 F and 108 NA for a weighted score of 18.6%. The false alarm rate for this system was only 3.4%, which is lower than that for the other spoken language and natural language systems on which we report in this paper. There is no obvious explanation for this. The simple hypothesis of better speech recognition in the Unisys-LL system will not suffice, because the BBN system has better speech recognition but the false alarm rate is higher than the Unisys-LL rate. In addition, the Unisys system's performance on the NL test tells us how the system would do given perfect speech recognition, and the false alarm rate there is around 8%. One possible hypothesis is that the bigram language model used in the Lincoln Labs system is in some sense more conservative than the language models used in the BBN and MIT systems and consequently prevents some of the inputs which might have led to an F in the natural language system from being recognized well enough for the natural language system to generate an F.

In this system, based on one input per utterance, we found that 59% of the inputs failed to receive a syntactic analysis (including compiled semantics, as discussed above) and 2% failed to receive a semantic analysis. No database call could be generated for 13% of the inputs and a call was made for the remaining 25% of the inputs.

Evaluation of the Natural Language System In [1] we reported on a technique for evaluation of the natural language component of our spoken language system, based on the question of how often did the natural language system do the right thing. If the reference answer for an utterance is found in the N-best, the right thing for the natural language system is to find the reference answer (or a semantic equivalent) in the N-best and give the right answer. The operational definition of doing the right thing, then, is for the system to receive a "T" on such inputs. On the other hand if the reference answer is not in the N-best the right thing for the system to do is to either find a semantic equivalent to the reference answer or to reject all inputs. Thus, doing the right thing in the case of no reference answer can be operationally defined as "T" + "NA".

	Reference in N-best	Reference not in N-best	Overall
Pundit right	54%	90%	84%
Pundit wrong	45%	10%	16%

Table 2: PUNDIT's performance on Class A (145 queries), depending on whether or not reference query occurred in N-best (N=16) from MIT-LCS SPREC.

	Reference in N-best	Reference not in N-best	Overall
Pundit right	69%	81%	73%
Pundit wrong	31%	19%	27%

Table 3: PUNDIT's performance on Class A (145 queries), depending on whether or not reference query occurred in N-best (N=16) from BBN SPREC.

Several interesting comparisons can be made based on tables 2 and 3. To begin with, it seems clear that the BBN N-best is better than the MIT N-best based on three quite distinct measures – first of all the speech recognition score is better (16.1% word error rate for BBN vs. 43.6% word error rate for MIT), secondly, the spoken language score (with the natural language system held constant) for Unisys-BBN is better than Unisys-MIT (39.3% for Unisys-BBN vs. 9.7% for Unisys-MIT) and thirdly, the reference answer occurred in MIT's top 16 candidates only 15% of the time vs. 65% of the time for the BBN N-best. Thus this experiment allows us

to ask the question of what effect does better speech recognition have on the interaction between speech recognition and natural language?

In the case where the reference answer is in the N-best, PUNDIT does much better with the BBN N-best. Since less search in the N-best is required with BBN data the reference answer or equivalent is likely to be found sooner, and consequently there will be fewer chances for PUNDIT to find a syntactically and semantically acceptable sentence in the N-best which differs crucially from what was uttered. On the other hand, PUNDIT actually does better with the poorer speech recognizer output from MIT when the reference answer is not in the N-best. We suspect that the poorer speech recognizer output is in some sense easier to reject; that is, it is more likely to seriously violate the syntactic and semantic constraints of English. If this is so then it is possible that a relatively accepting natural language system might work well with worse speech recognition outputs (because even a relatively accepting natural language system can reject very bad inputs), but with better speech recognizer output one might get good performance with a stricter natural language system. We plan to test this hypothesis in future research.

It is natural to ask why we should care about what to do with poorer speech recognizer output; one would think that we should use the best recognizer output possible. The answer is that many potential applications have requirements such as large vocabulary size which are somewhat at odds with high accuracy, consequently the best recognizer output available may nevertheless be relatively inaccurate. Thus it is important to have speech/natural language integration strategies which allow us to fine tune the interaction to compensate for less accurate speech recognition.

Optional Class A We used both the Unisys-MIT system and the Unisys-BBN system for this test. For both speech recognizers in this test of eleven utterances with verbal deletions we received two T's and zero F's for a weighted score of 18.2%. There is too little test data in this condition to draw reliable conclusions from the results.

Comparison of Spoken Language Systems

We believe coupling of a single natural language system with multiple speech recognition systems has the potential for being a very useful technique for comparing speech recognizers in a spoken language context. Of course speech recognizers can be compared on the basis of word and sentence accuracy, but we do not know how direct the mapping is between these measures of performance and spoken language performance. The most direct comparision for spoken language evaluation, then, is to define an experimental condition in which the systems to be compared differ only in the speech recognition component. Not only is this strategy useful for comparing system level measurements of performance of speech recognizers, but it is also useful for more fine grained analyses of the interaction between the speech recognition component and the natural language system.

Figure 3 shows the distribution of T's, F's and NA's for specific queries across the three systems.

Note that for 52 queries, or 36% of the total, the systems received the same score, although in no case did all three systems receive an "F". The largest difference among the three systems was in the number of cases where Unisys-BBN received a "T" but the other two systems received an "NA". This occurred for 31 queries.

Another interesting comparison is to look at the cases where Unisys-MIT and Unisys-BBN issued a call based on the first candidate in the N-best, since this corresponds to the one-best interface used in Unisys-LL. In Unisys-MIT twenty-seven calls were issued based on the first candidate, out of a total of 45 calls. Of the calls issued on the first candidate, 15 received a score of T and 12 received a score of F, for a weighted score of 2%. In Unisys-BBN the first candidate was selected from the N-best 70% of the time. 26 of these candidates resulted in scores of "F" and 42 resulted in a "T" for a weighted score of 11%.

Overall, the number of calls made was quite similar for the Unisys-LL and Unisys-MIT systems (25% of utterances for Unisys-LL and 30% for Unisys-MIT), but it was much higher for Unisys-BBN (67%). In all three systems most of the inputs were rejected by the syntax component (59% of all inputs for Unisys-LL, 74% of all inputs for Unisys-BBN and 85% of all inputs for Unisys-MIT). We can compare this to a baseline syntactic failure of 14% of inputs on the Unisys natural language test. (Note that since multiple inputs per utterance are possible with the N-best systems, the N-best vs. one-best systems are not strictly comparable.)

Speech Recognition Evaluations

Using speech recognition data from MIT, we submitted results for the Class A, Class D1, Class AO and Class D1O speech recognition tests, shown in tables 4, 5, 6, and 7.

As expected, we observed a higher error rate for the optional tests, which contained verbal deletions, and we also observed a wide range of performance across speakers. The comparison of D1 pairs and Class A speech recognition showed poorer word recognition in the D1 pairs than in the Class A test. An average 45.8% word error rate was observed for the Class A utterances compared to a 54.6% error rate for the D1 utterances. As tables 4 and 6 show, this was fairly consistent across speakers, except for speaker CJ. There are at least two hypotheses which may explain this higher error in context dependent spontaneous utterances. One hypothesis suggests that the higher error rate may be due in part to the presence of prosodic phenomena common in dialog such as destressing of "old" information. Because the specific dialog context affects the pronunciation of words corresponding to old and new information, the training data used so far may not provide a complete sample of how words are pronounced in a wide range of dialog contexts, consequently leading to poorer word recognition. Another hypothesis is based on the fact that the context-dependent sentences contain many references to flight numbers. Flight numbers may be difficult to recognize because there is very little opportunity for syntactic or semantic information to constrain which number was uttered.

CONCLUSIONS

In this paper we presented benchmark test results on natural language understanding, spoken language understanding

Speaker	Corr	Sub	Del	Ins	Err	S. Err
CE	56.0	33.3	10.6	5.1	49.1	95.0
CH	47.4	44.7	7.9	23.7	76.3	100.0
CI	45.6	46.8	7.6	24.0	78.4	100.0
CJ	75.8	18.3	5.9	3.1	27.3	84.6
CK	56.9	29.4	13.7	2.0	45.1	91.7
CL	75.0	22.5	2.5	9.7	34.7	84.6
CM	31.1	62.1	6.8	15.9	84.8	100.0
CO	74.1	19.1	6.8	8.6	34.6	100.0
CP	71.7	26.5	1.8	7.5	35.8	88.9
Average	63.5	30.3	6.2	9.3	45.8	91.2

Table 4: System Summary Percentages by Speaker for Class A

Speaker	Corr	Sub	Del	Ins	Err	S. Err
CE	59.5	40.5	0.0	16.2	56.8	100.0
CI	21.9	71.9	6.2	21.9	100.0	100.0
CJ	72.6	17.7	9.7	1.6	29.0	100.0
CM	42.7	50.7	6.7	21.3	78.7	100.0
Average	51.5	42.2	6.3	14.6	63.1	100.0

Table 5: System Summary Percentages by Speaker for Class AO

Speaker	Corr	Sub	Del	Ins	Err	S. Err
CH	40.0	53.3	6.7	13.3	73.3	100.0
CI	24.2	46.3	29.5	15.8	91.6	100.0
CJ	83.9	10.7	5.4	3.6	19.6	50.0
CK	48.0	39.9	12.2	2.7	54.7	100.0
CL	66.7	31.7	1.7	8.3	41.7	83.3
CO	54.7	27.3	18.0	3.1	48.4	94.4
CP	80.0	20.0	0.0	8.0	28.0	100.0
Average	52.0	34.1	13.9	6.6	54.6	91.4

Table 6: System Summary Percentages by Speaker for Class D1

Speaker	Corr	Sub	Del	Ins	Err	S. Err
CE	76.5	23.5	0.0	8.8	32.4	100.0
CM	34.8	65.2	0.0	26.1	91.3	100.0
Average	59.6	40.4	0.0	15.8	56.1	100.0

Table 7: System Summary Percentages by Speaker for Class D1O

and speech recognition. Our weighted score for the Class A natural language test was 48.3%, for the D1 pairs, 63.2%, for the Class AO test, 9.1% and for the DO test, -50%. We presented five benchmark tests of spoken language systems, Unisys-MIT on Class A, which received a weighted score of 9.7%, Unisys-MIT on Class AO, which received a weighted score of 18.2%, Unisys-LL on Class A, which received a weighted score of 18.6%, Unisys-BBN on Class A, which received a weighted score of 39.6%, and Unisys-BBN on Class AO, which received a weighted score of 18.2%. Finally, we presented speech recognition results using the Unisys natural language system as a filter on the N-best output of the MIT SUMMIT system.

The semantics enhancements to the natural language system are motivating us to revisit the tightly integrated architecture of semantics/pragmatics processing in our system, because with these enhancements, semantic information regarding a discourse entity can become available to the processing at a much later point than previously. Thus, pragmatic processing must be invoked at a later point to ensure that all relevant semantic information has been exploited.

The spoken language results are especially interesting, because we are now beginning to be able to look at the interactions of the natural language system with different speech recognizers, and see how to tune the natural language system to make the best use of the information available from the various speech recognizers. We believe that it is important to make these kinds of comparisons and we are planning to work with at least one other speech recognition system using the N-best interface. We also plan to begin exploring more tightly coupled systems using the stack decoder architecture ([9]).

REFERENCES

[1] Deborah A. Dahl, Lynette Hirschman, Lewis M. Norton, Marcia C. Linebarger, David Magerman, and Catherine N. Ball. Training and evaluation of spoken language understanding system. In *Proceedings of the DARPA Speech and Language Workshop*, Hidden Valley, PA, June 1990.

[2] Charles T. Hemphill, John J. Godfrey, and George R. Doddington. The ATIS spoken language systems pilot corpus. In *Proceedings of the DARPA Speech and Language Workshop*, Hidden Valley, PA, June 1990.

[3] L. Hirschman, M. Palmer, J. Dowding, D. Dahl, M. Linebarger, R. Passonneau, F.-M. Lang, C. Ball, and C. Weir. The PUNDIT natural-language processing system. In *AI Systems in Government Conf.* Computer Society of the IEEE, March 1989.

[4] F. Kubala, S. Austin, C. Barry, J. Makhoul, P. Placeway, and R. Schwartz. BYBLOS speech recognition benchmark results. In *Proceedings of the Darpa Speech and Natural Language Workshop*, Asilomar, CA, February 1991.

[5] F.-M. Lang and L. Hirschman. Improved portability and parsing through interactive acquisition of semantic information. In *Proc. of the Second Conf. on Applied Natural Language Processing*, Austin, TX, February 1988.

[6] Lewis M. Norton, Deborah A. Dahl, Donald P. McKay, Lynette Hirschman, Marcia C. Linebarger, David Magerman, and Catherine N. Ball. Management and evaluation of interactive dialog in the air travel domain. In *Proceedings of the DARPA Speech and Language Workshop*, Hidden Valley, PA, June 1990.

[7] Martha Palmer. *Semantic Processing for Finite Domains.* Cambridge University Press, Cambridge, England, 1990.

[8] D. B. Paul. New results with the Lincoln tied-mixture HMM CSR system. In *Proceedings of the DARPA Speech and Natural Language Workshop*, February 1991.

[9] Douglas B. Paul. A CSR-NL interface specification. In *Proceedings of the DARPA Speech and Natural Language Workshop*, 1989.

[10] V. Zue, J. Glass, D. Goddeau, Dave Goodine, Lynette Hirschman, H. Leung, M. Phillips, J. Polifroni, and S. Seneff. Development and preliminary evaluation of the MIT ATIS system. In *Proceedings of the DARPA Speech and Natural Language Workshop*, February 1991.

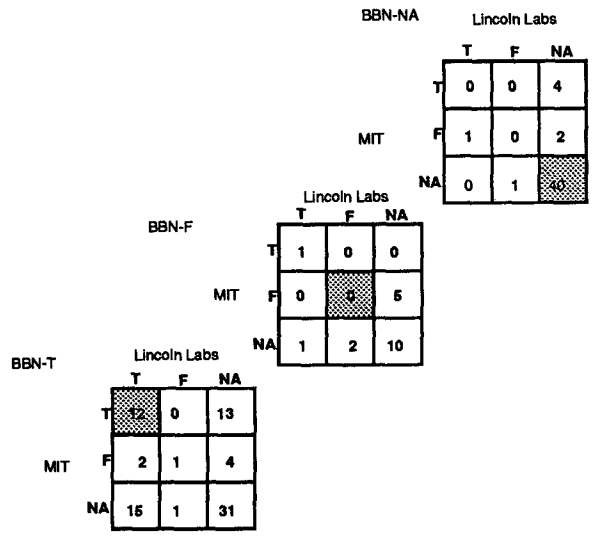

Figure 3: Comparison of results from three spoken language systems