Training and Evaluation of a Spoken Language Understanding System

**Deborah A. Dahl, Lynette Hirschman, Lewis M. Norton,
Marcia C. Linebarger, David Magerman,
Nghi Nguyen and Catherine N. Ball**

Unisys Defense Systems
Center for Advanced Information Technology
PO Box 517
Paoli, PA 19301

Introduction

This paper describes our results on a spoken language application for finding directions. The spoken language system consists of the MIT SUMMIT speech recognition system ([20]) loosely coupled to the UNISYS PUNDIT language understanding system ([9]) with SUMMIT providing the top N candidates (based on acoustic score) to the PUNDIT system. The direction finding capability is provided by an expert system which is also part of the MIT VOYAGER system [18]). [1]

One major goal in this research has been to understand issues of training vs. coverage in porting a language understanding system to a new domain. Specifically, we wished to determine how much data it takes to train a spoken language system to a given level of performance for a new domain. We can use the answer to this question in the process of designing data collection tasks to decide how much data to collect. We address a related question, that is, how to quantify the growth of a system as a function of training, in [12].

To explore the relationship of training to coverage, we have developed a methodology to measure coverage of unseen material as a function of training material. Using successive batches of new material, we assessed coverage on a batch of unseen material, then trained on this material until we reached a certain level of coverage, then repeated the experiment on a new batch of material. The system coverage seemed to level off at about 70% coverage of unseen data after 1000 sentences of training data.

A second goal was to develop a methodology for automatically tuning a broad-coverage grammar to a new application domain. This approach avoids repeating domain independent grammar development work over again for each new domain. To do this we developed a method for deriving a minimal grammar and a minimal lexicon from a corpus of training material. The application of this technique to the DIRECTION-FINDING corpus will be described in this paper. We then compared the coverage and performance of the minimal grammar and lexicon on a test set, and found that a two-fold decrease in parse time was achieved with only a small loss of coverage.

Our second major focus was on evaluation of specific algorithms, using the natural language system as a testbed. In particular, we compared the performance of two algorithms for reference resolution processing. Finally, our third major focus has been to evaluate the overall coverage and accuracy of the entire spoken language system. We did this using two test corpora collected at MIT ([17]), containing a total of 1015 utterances. The system was evaluated on the basis of the first utterance of the N-best output of SUMMIT accepted by PUNDIT, or the first candidate of the N-best if no utterance was accepted by PUNDIT. This paper reports results for word accuracy, sentence accuracy, application accuracy (generating an answer judged reasonable by naive evaluators), and finally false alarm rate (incorrect, incoherent or incomplete answers).

System Overview

The VOYAGER system has been described in detail elsewhere, ([18]) so we will only briefly describe it here. VOYAGER is a spoken language system for finding directions in Cambridge, Massachusetts. For example the user can ask questions about the locations of objects such as restaurants, universities, and hotels and distances between them. It provides output in the form of a map display as well as natural language. We have used the SUMMIT speech recognition system as well as the direction-finding expert system from VOYAGER in the system we are reporting on.

The architecture of the system which we report on here has also been largely described elsewhere ([1]), with the exception of the N-best processing, and so will only be summarized here. There are five major components of the system, the speech recognition system (SUMMIT), the dialog manager (VFE), the PUNDIT natural language processing system, the module which formats PUNDIT's

[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 for making the SUMMIT and VOYAGER systems available to us, and for providing us with the direction-finding data. We also wish to thank Mitch Marcus of the University of Pennsylvania for making the services of outside evaluators available to us, and Beatrice Santorini for coordinating the evaluation. Bill Scholz of the Unisys Center for Advanced Information Technology provided valuable assistance in the design and analysis of the black box evaluation task.

output for the direction finder (QTIP), and the direction finder itself. VFE takes SUMMIT's N-best output (computed using a word-pair grammar of perplexity 60), and sends it to PUNDIT for syntactic and semantic analysis. The first candidate which PUNDIT accepts is sent to QTIP, where it is formatted and sent to the direction finder. When the direction finder's English response is returned to VFE, it is sent to PUNDIT as well, so that the information from the direction finder's response can be processed and incorporated into the discourse context. This feature allows the user to refer to things that the direction finder has mentioned, and in general allows the user and the expert system to engage in a dialog. The PUNDIT natural language processing system ([9]) is implemented in Prolog and consists of a top-down backtracking parser ([10]), a semantic interpreter ([13]), and a pragmatics component ([5]). The system uses a semantic-net based knowledge representation system ([6]).

Training

Training Data and Coverage

In order to measure the effect of training on coverage, we developed a standardized training technique. We began by training the system on a set of 176 development sentences to a level of 96% percent application accuracy. We then ran a batch of 300 previously unseen sentences through the system and measured accuracy. This was followed by a development stage where we trained the system to about 80% accuracy. Then the system was given a new set of 300 unseen sentences. This cycle of testing and development was repeated for approximately 1000 sentences, and we observed a leveling off of the performance of the system on unseen data at approximately 70%. The growth of coverage is illustrated in Figure 1. The cold run coverage is coverage measured on each batch of sentences before any development had taken place. Development coverage was the coverage after the system was developed for that batch of sentences, and final coverage was the increased coverage that was achieved after development on later batches of sentences.

Grammar Pruning Experiments

Use of tight syntactic and semantic constraints is an important source of constraint in a spoken language understanding system. There are two approaches to constructing a tight grammar for a given corpus of training material. One approach is to build the grammar incrementally, based on the observed training data, as in TINA ([14]). This approach has the disadvantage of constructing a basic grammar of English over again for each domain. The other approach is to prune a general grammar of English to cover only those constructions seen in the training data. This approach has the advantage of making available a 'library' of constructions (that is, the full grammar) which can easily be added to the system when additional data indicates a need for them. In both cases, the coverage of the grammar will directly reflect the amount of training data seen.

We have developed a technique for pruning our general English grammar, based on supervised training. We make use of the fact that PUNDIT provides a detailed parse tree, reflecting the set of BNF definitions used in parsing the sentence. The parse tree also contains each word labeled by part of speech. Given a corpus of training sentences with their correct parses, a program can identify those constructions used to obtain that parse and can extract the associated rules from the general grammar. Similarly, it can identify how each word is actually used in the training data and extract a minimal lexical definition reflecting the word's usage in context.

Using these techniques, we performed a small set of experiments on the effects of pruning both the grammar and the lexicon. There are several ways to analyze how pruning affects overall system behavior. We can look for reduction in perplexity; however, given the heavily context-dependent nature of our grammar, the effects of pruning seemed quite small (10-15% reduction). We also looked at the effect of pruning on overall system performance. Given a grammar based on some 500 training sentences, we observed a two-fold speed-up when the same sentences were run using the pruned grammar and lexicon (reducing the average time to correct parse from 4.2 sec to 2.1 seconds, on a Sun 3-60). We also looked at the relation between coverage and amount of training data. Our tests indicated that we did not lose much coverage by pruning the grammar (-3 % after training on 226 sentences and -2 % after training on 526 sentences). We lost more coverage from pruning the lexicon (-33 % after training on 226 sentences, and -7 % after training on 526 sentences), but this was largely to the "unknown" word problem, more than to pruning away needed meanings.

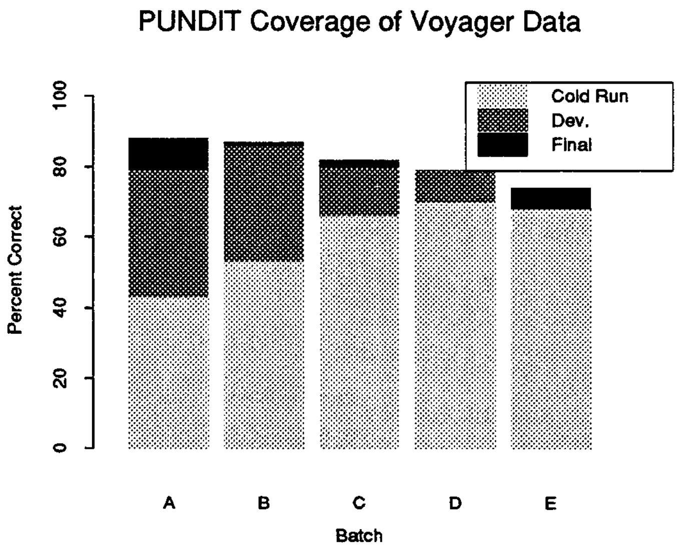

Figure 1: Coverage of VOYAGER data on 5 successive sets of 300 sentences

These experiments, though limited, indicate that grammar and lexicon pruning may offer significant reductions in processing time with only small losses in coverage. Furthermore, although we have not yet experimented with the pruned grammar on spoken input, we expect that the pruned grammar will improve the ability of PUNDIT to reject ungrammatical candidates from a speech recognizer.

Evaluation

We have explored several new approaches to evaluation, including using the natural language system as a tool for comparing specific algorithms, and subjective black box evaluation of accuracy as well as standard word and sentence accuracy measurements.

Reference Resolution Algorithms

Using the language understanding system to compare algorithms gives a very tightly controlled comparison of two or more specific approaches to a problem, and is appropriate when the algorithms of interest are modular, as they are in this case.

We investigated the performance of two reference resolution algorithms, using PUNDIT as a testbed. The data consisted of a set of 68 discourses (the set of discourses with inter-sentential anaphoric references) taken from the 1000 direction finding utterances which had been processed by PUNDIT. The reference resolution algorithms were both variations of the focusing/centering approach ([7],[8], [15])). We compared object and subject preference. Object preference means that the direct object from the previous sentence is preferred over the subject as the referent of a pronoun in the current sentence. Subject preference means that the subject is preferred. Both approaches have been advocated in the literature, ([2],[3],[11]) but have not been tested using naturally occurring data. We ran the set of 68 discourses and tabulated accuracy of reference resolution for each pronoun. No significant difference in accuracy was found. Accuracy was near 100 % in both cases. The amount of time spent by the system performing reference resolution was also measured and was found not to differ significantly in each condition. These findings suggest that both algorithms perform equally well in this domain. This reflects the fact that there are very few instances where both a subject and an object are competing candidates for a referent in this data. Additional details of this investigation are reported in [4].

System Evaluation

We used a subjective black box measure and word and sentence accuracy to evaluate the the system as a whole. Subjective black box evaluation allows us to evaluate performance on queries which have an unspecified number of acceptable answers. For example, we can evaluate vague queries, which require a clarification dialog to elicit more information before they can be submitted to the application back end. We can also ask questions of human judges that cannot be asked of an automatic evaluation technique, such as whether an answer was partially correct, or whether an error message was helpful or not. Although there is always a question of reliability when human judges are used, when judges are provided with clear and explicit instructions, human judgements can be quite reliable.

Methodology

The system was evaluated at the level of word, sentence and application accuracy. Word and sentence level accuracy were measured using the NIST evaluation software. In order to evaluate the system at the application level, we designed a black box evaluation task, using human evaluators to score each interchange of a dialog between the system and a user. The evaluators were five students at the University of Pennsylvania and one Unisys employee who was not a system developer.

This evaluation task was similar to one reported by MIT ([19]) in that the evaluators were asked to categorize both the queries and the responses. The queries were categorized as to whether they were appropriate or inappropriate, given the capabilities of the application. They were also categorized on a three point scale of clarity, where 1 represents a clear and fluent query, 2 a partially garbled or badly stated query, and 3 represents a query that is partially or entirely uninterpretable. The responses were categorized first as to whether they were answers or error messages. The answers were then subdivided into 'correct', 'partially correct', and 'incorrect or misleading'. The error messages were categorized as either 'helpful', 'unhelpful', or 'incorrect or misleading'. The logs from Test Set 1, containing the original orthographic transcription of the query plus the system's response, were scored by three judges. Due to time constraints, the logs of Test Set 2, also containing the reference utterance and the system's response, were scored by only one judge. The logs were presented to the judges via an interactive program which displays a query/response interchange (including intermediate clarification dialog) and elicits responses for each category.

Results

Word accuracy with PUNDIT as a filter was computed on the basis of the first candidate accepted by the syntactic and semantic components of PUNDIT, or if no candidate was accepted, on the first candidate. Word accuracy without PUNDIT was computed on the basis of the first candidate of the N-best (i.e., the candidate with the highest acoustic score).

Table 1 shows the results on Test Set 1 for word, sentence, and application accuracy. Table 2 displays the results for Test Set 2. For the purposes of application accuracy, 'correct response' means either a correct answer or a helpful error message providing a meaningful diagnosis of a query falling outside the system. 'False

	Word Accuracy		Sentence	Application Accuracy	
	Correct	Error	Accuracy	Correct	False Alarm
W/ PUNDIT as filter	80.0	26.1	34.3	45.7	10.7
W/O PUNDIT as filter	76.4	31.1	20.6	22.0	9.3

Table 1: Word, sentence and application accuracy with and without PUNDIT filter for Test Set 1 (11 speakers, 519 query/response pairs) (PUNDIT interpreting the utterance in all cases)

alarms' include partially correct responses, incorrect responses, and incorrect error messages. The natural language component in our current system plays two roles; that is, as filter for the speech recognizer and as interpreter of recognized utterances. For this reason, we have distinguished these roles in the tables. Using PUNDIT as a filter means that the utterance recognized by the spoken language system is the candidate accepted by PUNDIT. Not using PUNDIT as a filter means that the utterance recognized by the spoken language system is the first candidate of the N-best. Using PUNDIT to interpret the utterance means that the candidate selected from the N-best (by whatever means) is sent to PUNDIT for interpretation and translation into function calls to the direction finding expert system.

Analysis

Using human judges for the black box evaluation requires assessing their reliability. Judges must be able to agree on their classifications or this approach will not be useful. We measured the reliability of the judges in the black box evaluation by using an analysis of variance technique described in [16]. For the judgement of most interest, that is, whether the answer was correct, partially correct, or incorrect, the mean reliability averaged over speakers was .78 with a standard deviation of .09 for the Test Set 1. Since we had only one judge for Test Set 2, we do not have reliability measurements for that data.

This reliability score can be interpreted as saying that if a new set of judges did these tasks we would expect the correlation between the new judgements and the old judgments to have this value. The high reliability of the correctness judgement is not surprising, since this is a fairly objective judgement. The other, more subjective, judgements which we asked the evaluators to make were less reliable than correctness. For example, the reliability of the fluency judgements on the test data was only .34. We believe that this could be improved through more explicit instructions and some additional training for the evaluators, although it may be that the fluency judgement is of only marginal interest anyway. Disagreements among the judges were mediated by an expert judge who was familiar with the evaluation task, but was not a system developer.

Tables 1 and 2 show that, while PUNDIT as a filter did not provide a large improvement in word accuracy, it did provide a fairly large improvement in sentence accuracy, and it roughly doubled application accuracy. Application accuracy went from 22.0 to 45.7 percent for Test Set 1 and from 28.0 to 51.6 percent for Test Set 2. This improvement results from PUNDIT'S ability to reject uninterpretable candidates in the set of N-best candidates, so that only meaningful candidates are considered for interpretation. It is also interesting to note that word accuracy and application accuracy do not covary completely. There are two reasons for this. First, it is possible for a candidate that is semantically equivalent to the reference answer to be accepted by the natural language system and for the answer to be judged correct. For example, this would be the case if the reference utterance was *How do I get to the nearest bank?* but the natural language system accepted *How would I get to the nearest bank?*. The upshot of this situation is a correct score for application accuracy but a lower score for word accuracy. On the other hand, it is possible for the reference candidate to be accepted by the natural language system, but to then be misunderstood and consequently give an incorrect response. This means that word accuracy is good even though application accuracy is bad.

	Reference in N-best	Reference not in N-best	Overall
Pundit right	76.0	91.9	83.6
Pundit wrong	24.0	9.1	17.4

Table 3: PUNDIT's performance on Test Set 1 (11 speakers, 519 query/response pairs), depending on whether or not reference query occurred in N-best (N=40).

	Reference in N-best	Reference not in N-best	Overall
Pundit right	83.5	86.1	84.7
Pundit wrong	16.5	13.9	16.3

Table 4: PUNDIT's performance on Test Set 2 (10 speakers, 496 query/response pairs), depending on whether or not reference query occurred in N-best (N=100).

Over and above these summary scores, we wished to investigate the performance of PUNDIT depending on the state of affairs in the N-best. For example, we looked at what PUNDIT did when the reference query occurrs in the N-best and when it does not. If the reference query

	Word Accuracy		Sentence	Application Accuracy	
	Correct	Error	Accuracy	Correct	False Alarm
W/ PUNDIT as filter	77.5	29.1	33.7	51.6	9.3
W/O PUNDIT as filter	74.1	33.4	20.8	28.0	2.2

Table 2: Word, sentence and application accuracy with and without PUNDIT filter for Test Set 2 (10 speakers, 496 query/response pairs), PUNDIT interpreting the utterance in all cases)

occurs in the N-best then the right thing for the natural language system to do is to find it, or find a semantically equivalent candidate, and give a correct answer. On the other hand, if the reference query is not in the N-best, then the right thing to do is either to find a semantically equivalent candidate and give a correct answer, or to reject all candidates. Table 3 shows how often PUNDIT did the right thing, by these criteria, for Test Set 1 with N=40, and Table 4 shows similar results for Test Set 2 with N=100.

We were also interested in looking at the performance of the system as a function of the location of the reference answer in the N-best. This bears on the question of what the optimal setting is for N. Intuitively, looking at more candidates increases the probability that the reference utterance or a semantic equivalent will be found, but at the same time it increases the probability of a false alarm, with the natural language system finding an acceptable candidate that differs semantically from the reference utterance in crucial ways. If we could quantify the relationships among N, the rate of correct responses, and the false alarm rate, it would give us a technique for setting N for optimal accuracy of the spoken language system, given a particular speech recognizer and a particular language understanding component.[2] In Figure 2, we show the cumulative correct answers and the cumulative false alarms as a function of the location of the reference utterance in the direction finding data from Test Set 1 and Test Set 2. In order to determine the optimal setting of N for the SUMMIT/PUNDIT system, we looked at the difference between the cumulative correct responses and false alarms as a function of the location of the reference utterance in the N-best. Figure 3 shows this difference, assuming that correct responses and false alarms have an equal weighting. Obviously they do not have an equal weighting in general. In fact, in most applications false alarms should probably be heavily penalized. If we weight the cost of a false alarm at three times the benefit of a correct response, then the optimal performance of this system is obtained with N = 10 for Test Set 1 and N = 11 for Test Set 2. That is, if the answer is not found in the top 10-11 candidates, the cost of false alarms exceeds the benefit of increased correct responses.

Conclusions

This paper has described several approaches to training and evaluation of spoken language systems. In the area of training we described an approach to measuring the performance of the system on previously unseen data as increasing amounts of training data are used. This experiment demonstrated that a level of 70 % coverage of unseen data was reached after the system was trained on 1000 sentences. We also described how a general, broad coverage grammar and lexicon can be automatically pruned to fit a specific application, thus saving the effort involved in building grammars from scratch for each application.

We also described several approaches to evaluation. We used the system as a tool to evaluate alternative algorithms for reference resolution, and we also evaluated the entire system on word, sentence, and application accuracy. Application accuracy was evaluated using a black box technique with evaluators who were not system developers. We found that the evaluators used in this task were relatively reliable, and we expect that improvements in training and instructions would improve the reliability. Overall application accuracy for the test data was 51.6 %. Separating the performance of PUNDIT from that of the entire spoken language system, we found that PUNDIT did the "right thing" 84.7 % of the time for the test data. Finally we demonstrated a new technique for determining the optimal setting of N for the N-best output from the speech recognizer in a loosely coupled system, for a given speech recognizer, language understanding system, and application.

References

[1] Catherine N. Ball, Deborah Dahl, Lewis M. Norton, Lynette Hirschman, Carl Weir, and Marcia Linebarger. Answers and questions: Processing messages and queries. In *Proceedings of the DARPA Speech and Natural Language Workshop*, Cape Cod, MA, October 1989.

[2] Susan E. Brennan, Marilyn W. Friedman, and Carl J. Pollard. A centering approach to pronouns. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pages 155–162, Stanford, CA, 1987.

[3] Deborah A. Dahl. Focusing and reference resolution in PUNDIT. In *Proceedings of the 5th National*

[2] This leaves out another important component of optimal N, of course, which is time, since a larger N would normally be expected to result in increased processing time.

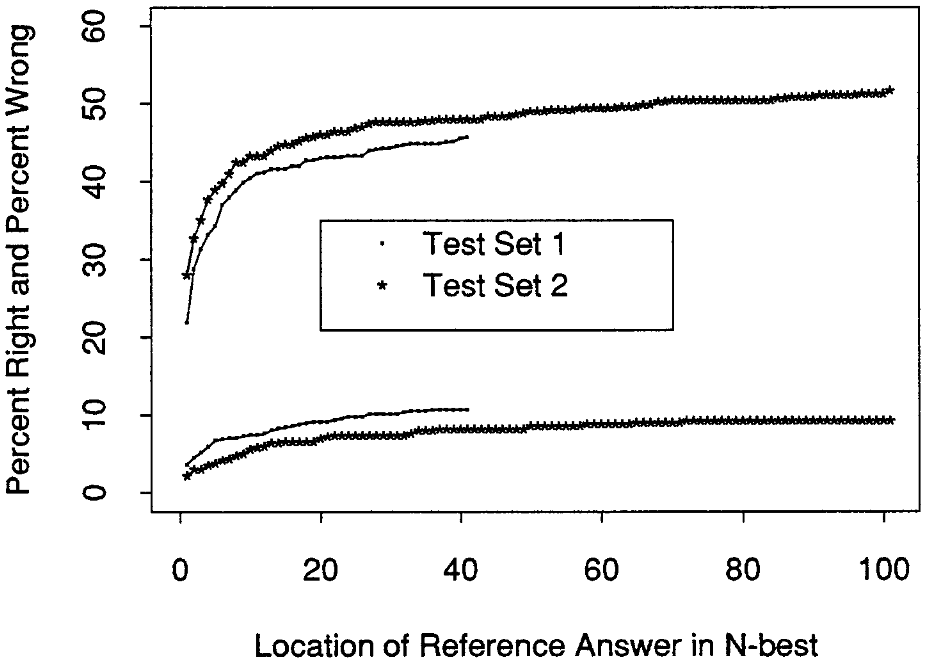

Figure 2: Percent right and percent false alarm vs. location of reference query in N-best, for Test Set 1 and Test Set 2

Conference on Artificial Intelligence, Philadelphia, PA, August 1986.

[4] Deborah A. Dahl. Evaluation of pragmatics processing in a direction finding domain. In *Proceedings of the Fifth Rocky Mountain Conference on Artificial Intelligence*, Las Cruces, New Mexico, 1990.

[5] Deborah A. Dahl and Catherine N. Ball. Reference resolution in PUNDIT. In P. Saint-Dizier and S. Szpakowicz, editors, *Logic and logic grammars for language processing.* Ellis Horwood Limited, in press.

[6] Michael Freeman, Lynette Hirschman, Donald McKay, and Martha Palmer. KNET: A logic-based associative network framework for expert systems. Technical Memo 12, SDC—A Burroughs Company, P.O. Box 517, Paoli, PA 19301, September 1983.

[7] Barbara Grosz, Aravind K. Joshi, and Scott Weinstein. Providing a unified account of definite noun phrases in discourse. *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pages 44–50, 1983.

[8] Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. Towards a computational theory of discourse interpretation. unpublished mss., 1986.

[9] Lynette Hirschman, Martha Palmer, John Dowding, Deborah Dahl, Marcia Linebarger, Rebecca Passonneau, François-Michel Lang, Catherine Ball, and Carl Weir. The PUNDIT natural-language processing system. In *AI Systems in Government Conference.* Computer Society of the IEEE, March 1989.

[10] Lynette Hirschman and Karl Puder. Restriction grammar in prolog. In *Proceedings of the First International Logic Programming Conference*, pages 85–90. Association pour la Diffusion et le Developpement de Prolog, Marseilles, 1982.

[11] Megumi Kameyama. *Zero Anaphora: The Case of Japanese.* PhD thesis, Stanford University, Stanford, CA, 1985.

[12] 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.

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

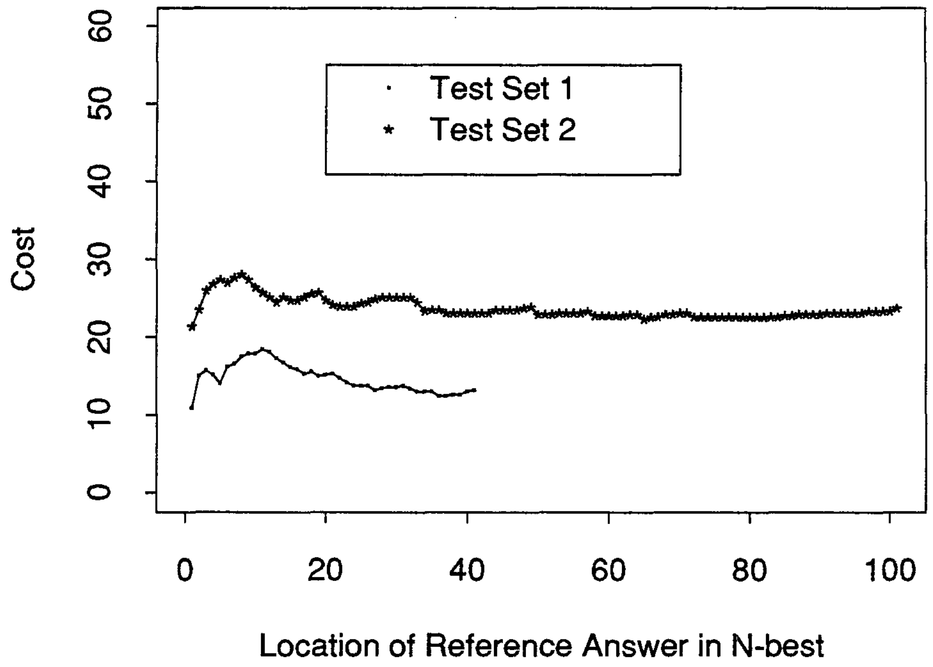

Figure 3: Percent right - percent false alarm vs. location of reference query in N-best, for Test Set 1 and Test Set 2

[14] Stephanie Seneff. Tina: a probabilistic syntactic parser for speech understanding systems. In *Proceedings of the First DARPA Speech and Natural Language Workshop*, Philadelphia, PA, February 1989.

[15] C.L. Sidner. *Towards a computational theory of definite anaphora comprehension in English discourse.* PhD thesis, MIT, 1979.

[16] B. J. Winer. *Statistical Principles in Experimental Design.* McGraw-Hill Book Company, New York, 1971.

[17] Victor Zue, Nancy Daly, James Glass, David Goodine, Hong Leung, Michael Phillips, Joseph Polifroni, Stephanie Seneff, and Michal Soclof. The collection and preliminary analysis of a spontaneous speech database. In *Proceedings of the DARPA Speech and Natural Language Workshop*, Cape Cod, MA, October 1989.

[18] Victor Zue, James Glass, David Goodine, Hong Leung, Michael Phillips, Joseph Polifroni, and Stephanie Seneff. The VOYAGER speech understanding system: A progress report. In *Proceedings of the DARPA Speech and Natural Language Workshop*, Cape Cod, MA, October 1989.

[19] Victor Zue, James Glass, David Goodine, Hong Leung, Michael Phillips, Joseph Polifroni, and Stephanie Seneff. Preliminary evaluation of the voyager spoken language system. In *Proceedings of the DARPA Speech and Natural Language Workshop*, Cape Cod, MA, October 1989.

[20] Victor Zue, James Glass, Michael Phillips, and Stephanie Seneff. The MIT SUMMIT speech recognition system: A progress report. In *Proceedings of the First DARPA Speech and Natural Language Workshop*, Philadelphia, PA, February 1989.