Evaluation Methodology for a Telephone-Based Conversational System

Joseph Polifroni, Stephanie Seneff, James Glass, and Timothy J. Hazen

Spoken Language Systems Group, Laboratory for Computer Science
Massachusetts Institute of Technology, Cambridge, MA 02139 USA
http://www.sls.lcs.mit.edu

Abstract

This paper describes a suite of metrics that we use to evaluate our JUPITER system which provides world-wide weather information over the telephone. Since May, 1997, we have made the system available to the general public via a toll-free number and have collected approximately 35,000 utterances to date. These data have proven invaluable for system development and evaluation. Because JUPITER makes use of many component spoken language technologies, we evaluate each as a stand-alone system. However, JUPITER must also be evaluated as to how it accomplishes its main goal, that of understanding user queries and providing an informative response. This requires an evaluation that takes into account how these various technologies fit together in an end-to-end fashion. We will describe an automated end-to-end evaluation system we have implemented and compare it to a subjective evaluation conducted by hand.

1. Introduction

Over the past several years, we have become increasingly interested in displayless systems. We are currently developing a system called JUPITER (Zue, et al., 1997), which allows a user to access and receive on-line weather information over the phone and in multiple languages. JUPITER utilizes the client-server architecture of GALAXY (Goddeau, 1994), and it specializes in world-wide weather-specific information obtained from a variety of sites accessible via the internet. It can give a weather report for a particular day or several days, and answer specific questions about weather phenomena such as temperature, wind speed, precipitation, pressure, humidity, sunrise time, etc. JUPITER serves as a testbed for several important areas that have surfaced on our research agenda, including displayless interaction, virtual browsing, information on demand, and translingual content management. The system currently has weather information for several hundred cities, mostly within the United States, but also selected major cities world-wide. The information is currently available in English, but we have begun work on a multilingual version of JUPITER, currently concentrating on Spanish and Mandarin Chinese.

Since late April 1997, we have made the system available to the general public via a toll free number. With remarkably little effort in maintaining the on-line system, we have collected, and are continually adding to, a corpus of spontaneous, goal-directed speech that serves the research needs of each component of the system. As of this writing, we have collected more than 35,000 utterances from over 6,500 callers. These data have proven invaluable for system development and evaluation. The purpose of this paper is to share our experience in development of a suite of evaluation methodologies and metrics for conversational systems, realizing that the process is still evolving over time. We will begin by describing our motivation for pursuing the work described in this paper and then briefly describe the process by which we continue to collect the JUPITER corpus. After providing some details on the JUPITER corpus itself, we will describe the various evaluation methodologies we have developed to measure progress and assess the quality of the various components of our system. We will conclude with a discussion of the lessons we have learned from JUPITER.

2. Motivation

There are two ways to evaluate a spoken language system. In one evaluation method, system behavior is judged by examining each query/response pair. Component evaluation, on the other hand, examines the behavior of each part of the system to see how well each performs separately. We employ both types of evaluation in JUPITER, with the overall goal of understanding system behavior as thoroughly as possible.

The main components of the JUPITER system are the speech recognizer SUMMIT (Glass et al., 1996), the natural language parser TINA (Seneff, 1992), the natural language generation component GENESIS (Glass et al. 1994), and the JUPITER domain server, which queries the database for the information that the caller requested and composes system responses. Each component is in a state of active development; furthermore, the components interact with each other in the on-line system, making it inaccurate or even impossible to evaluate them individually. When changes are made to the system, it is critical to be able to assess the impact of those changes on system performance.

Because our main goal in developing JUPITER was to provide useful information to users, we have been most concerned with developing evaluation metrics that help us understand how well JUPITER understands. In addition to metrics to assess speech recognition and parse coverage, we will describe below an automated mechanism to evaluate the understanding of the user query, and compare this automated mechanism to a subjective evaluation done by hand.

We have had some prior experience with understanding evaluation in the ATIS (Air Travel Information Service) domain (Hirschman et al., 1993), the common task for DARPA spoken language evaluation from 1990 to 1994. In the ATIS system, users were asked to solve experimental scenarios by speaking to a system. Utterances were collected by several sites, transcribed at each site, and then sent to SRI for annotation in the CAS (Common Answer Specification) format. Researchers at SRI evaluated each utterance and, for those deemed answerable, prepared ancillary files containing a meaning representation, in a formal language known as *NLParse*[1]. A set of database "tuples" were then obtained from a common static SQL database by further automatic processing of the *NLParse* representation. Each participating site in the testing process received training data comprised of all of these files, as well as software provided by NIST for scoring hypothesized database tuples against the reference tuples supplied by SRI.

The entire process was quite time-consuming, as was the process of agreeing to and maintaining a set of "Principles of Interpretation" such that each site could know exactly how to interpret queries about such things as direct or "red-eye" flights. When these principles were either inconclusive or missing for certain types of queries, the process of adjudication, whereby a site argued for its interpretation for a particular query, could involve a great deal of time and effort.

Although the ATIS experience was useful, it did not assess something we consider a very important part of a spoken language system: how the system and the user interact (Price et al., 1992). Our philosophy with JUPITER has been to be as helpful as possible, sometimes providing information that the user did not specifically ask for, in an attempt to both answer the query and inform the user about the capabilities of the system. For example, if the user asks about humidity in a particular city and JUPITER does not have that information, the system will offer the percent humidity for a nearby city. Although a "correct" response might be "I'm sorry, I don't have the information you requested" we prefer a response such as "I'm sorry, I don't have humidity information for Sacramento. The only city in California for which I have humidity values is San Francisco. In San Franscisco the humidity is 60%".

Another drawback of the CAS evaluation paradigm was that we did not have a set of human-annotated answer files to compare our system against. JUPITER is under continual development, with capabilities and functionality added as new weather resources are found or user queries inspire. Furthermore, JUPITER's database of weather information changes significantly several times a day. Without requiring that the database nor the system capabilities be frozen, we wished to explore an automated methodology for evaluating understanding, based on the actual *meaning* of the input utterance, instead of the response as generated from the database. We feel that this meaning representation is an accurate way of assessing our system's linguistic competence within the domain.

3. The Jupiter Corpus

3.1. Corpus Collection/Transcription

The first data we collected within the JUPITER system was read speech, which we solicited, in groups of 50 utterances, from members of the Spoken Language Systems Group and students in a course on automatic speech recognition (Hurley et al., 1996). We then conducted a round of wizard-of-Oz data collection, where subjects were asked to solve several different scenarios, with a transcriber serving the role of speech recognizer. This produced a considerable number of utterances per user; however, the wizard data were time-consuming to collect. Eventually, we felt we had sufficiently robust recognition and understanding capabilities to publicize the toll-free number and have users interact with the system.

We are currently averaging approximately 23 calls per day to JUPITER. In order to ensure that these data are ready to use as quickly as possible for both speech recognition and natural language development, we have been processing incoming data on a daily basis. Every morning a script automatically sends email containing a list of the previous days calls to our group secretary, who then manually transcribes the utterances, usually over the course of the following day. The transcribed calls are then bundled into sets containing approximately 500 utterances and are added to the training corpus as they become available (with selected sets periodically set aside for testing).

Over the past year, we have continued to refine our transcription tool which was originally developed for orthographic transcription of read speech. We used a Tcl/Tk interface to provide an editable window where the transcriber can listen to utterances and correct existing transcriptions, adding specialized markings for noise, partial words etc. The initial transcription for each utterance is the orthography hypothesized by the recognizer during the call. The transcriber is also asked to identify the talker as male, female, or child using the transcription tool.

The transcription tool uses a lexicon to check the quality of orthographic transcriptions. This feature is useful for finding typographical errors before they are saved. If a word does not appear in the lexicon, the transcriber is warned and given the option of either

[1] *NLParse* is a proprietary system from Texas Instruments, which the DARPA community was allowed to use for the ATIS task.

User characteristic	Percentage of data
Male	70.7
Female	20.8
Children	8.5
Heavily accented	10.3
Foreign language	.1

Table 1: A profile of the user population for JUPITER.

Signal characteristic	Percentage of data
Speaker phone	5
Cellular/car phone	.5
Regular handset	94.5
Noise	11
Filled pauses	7
Partial words	6

Table 2: A breakdown of line characteristics and disfluent speech from the JUPITER corpus.

adding the word to the lexicon or changing it in the orthography file. The use of the lexicon also allows us to monitor the growth of new words in the domain.

3.2. Corpus Analysis

Table 1 provides a snapshot profile of the user population for our toll-free line. Just over 70% of users were male speakers, with females comprising approximately 21% of the data, and children the remainder. A portion of the data was from non-native speakers, although the system performs adequately on speakers whose dialect or accent does not differ too much from general American English. Callers with strong accents however, have been thus far excluded from training sets for both the speech recognition and natural language components. These data constituted approximately 10% of the data, and will be useful for future study. A very small fraction (0.1%) of the utterances included talkers speaking in a foreign language (e.g., Spanish, French, German, Chinese).

Our transcriber also marks utterances for various disfluencies and/or signal characteristics, a breakdown of which can be seen in Table 2. The signal quality of the data varied substantially depending on the handset, line conditions, and background noise. Although an underestimate, speaker phones were clearly used in approximately 5% of the calls due to the presence of multiple talkers in an utterance. Only a small fraction of the data (0.5%) was estimated to be from cellular or car-phones.

Over 11% of the data contained significant noise. About half of this noise was due to cross-talk from other speakers, while the other half was due to non-speech noises. The most common identifiable non-speech noise was due to the user hanging up the phone at the end of a recording (e.g., after saying good bye). Other distinguishable sources of noise included (in descending order of occurence) television, music, phone rings, touch tones etc. These data are excluded from training of the speech recognition component, although cleaned up orthographies are used for natural language training.

There were a number of spontaneous speech effects present in the recorded data. Over 6% of the data included filled pauses (e.g., uh, um, etc.) which were explicitly modeled as words in the recognizer, since they had consistent pronunciations, and seemed to occur in predictable places in utterances (filled pauses were removed from the sentence hypotheses sent to the natural language component). Utterances contained partial words in another 6% of the data, although approximately two thirds of these were due to clipping at the beginning or end of an utterance. The remaining artifacts were contained in less than 2% of the data and included phenomena such as (in descending order of occurence) laughter, throat clearing, mumbling, shouting, coughing, breathing, sighing, sneezing, etc.

The data from foreign speakers, as well as the data containing cross-talk or other noises, clipped speech or other spontaneous phenomenon (excluding filled pauses), collectively accounted for approximately one quarter of all recorded data. To date, these data have not been used for training the speech recognizer, although all utterances are included in the results we report on evaluation, unless otherwise noted. In practice, the system often performs well on these data during an actual call. The reason we have avoided these data is due to the concern that the recognizer will produce poor alignments during training which will ultimately contaminate the acoustic-phonetic models. We hope to test this hypothesis and examine these data more closely in the near future.

4. Evaluation

Figure 1 shows the set of metrics we use to evaluate JUPITER, along with the system components that are tested with each metric. In the following sections, we discuss each evaluation metric to present an overall picture of how we keep current on system performance.

4.1. Speech Recognition

The vocabulary used by JUPITER has evolved over the course of the year as periodic analyses were made of the growing corpus. The current vocabulary consists of 1893 words, and contains 638 cities, and 166 countries. A class bigram language model is used in the forward Viterbi search, while a class trigram model is used in the backwards A^* search to produce the 10-best outputs for the natural language component. When tested on a 1445 utterance test set the word-class bigram and trigram had perplexities of 16.6 and 15.4, respectively.

We periodically train a new recognizer from a continuously growing corpus of acoustic data, and we need to evaluate each new recognizer carefully before inserting it into our on-line system. Figure 2 shows the results of a performance evaluation from the spring of

Evaluation method	Components tested
Word/sentence accuracy	Recognizer
Parse coverage	Parser
Paraphrase comparisons	Content understanding, Generation
Understanding score	Recognizer, Parse, Discourse
Static database assessment	Understanding, Discourse, Dialogue, Database access, Generation
Logfile evaluation	Recognition, Understanding, Discourse, Dialogue, Database access, Generation

Figure 1: Evaluation metrics used in the Jupiter system and their corresponding components.

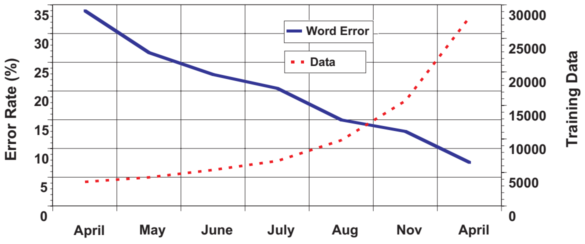

Figure 2: Word error rate on within-vocabulary utterances plotted over the first year of JUPITER's deployment.

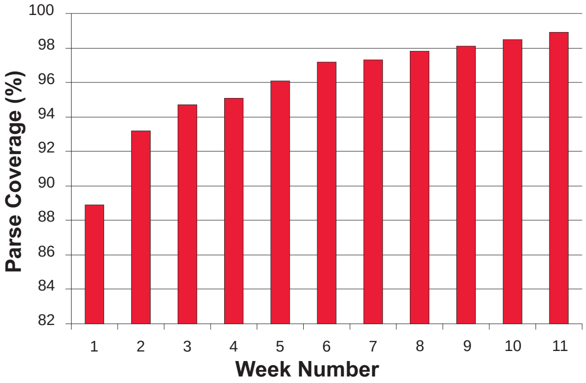

Figure 3: Parse coverage over the first eleven weeks of grammar development for National Weather Service weather reports.

1998, plotting the performance of the recognizer on a set of data recorded in the fall of 1997. All testing has been performed on the subset of collected data considered to be within domain, and excludes utterances with out-of-vocabulary words, clipped speech, cross-talk, and other kinds of noise. The within-domain utterances typically correspond to 70 to 75% of the recorded data. Note that the understanding component is often able to correctly answer the excluded utterances in the on-line system.

As shown in Figure 2, the performance has consistently improved over time as the recognizers have been able to incorporate more sophisticated language and acoustic models due to increased amounts of training data. At the end of April, 1997, the recognizer was trained primarily on read speech, and wizard-based spontaneous speech. This laboratory trained system had word-error rates of 7% on read-speech and 10% on spontaneous speech collected from group members. However, the error rates initially tripled on incoming data from real users. Over the course of the year both word and sentence error rates have been reduced by a factor of three; word error rate is now 7.6% and sentence error, 21.5%.

4.2. Parsing

In the JUPITER system, the output of the parsing step is a meaning representation that we call a "semantic frame", an example of which is shown in Figure 4. Parsing for JUPITER involves both the weather reports themselves and user queries. Weather reports are updated several times a day from various sources on the Web and over the internet. An automatic procedure parses the data into semantic frames, and a second process sorts them into categories based on the meaning. Each weather report is first converted to an indexed list of semantic frames, one for each sentence. The indices are then entered into the relational database under appropriate topicalized categories.

After less than three months of operation, JUPITER had achieved a very high parse coverage of incoming weather reports, currently hovering in the 99% range. There always seem to be a few sentences that fall outside of the grammar's domain, but any sentences that fail to parse can be rephrased by the system developer to recover an equivalent meaning. The grammar can later be expanded to encompass a broader base. By requiring a full parse and hand-editing sentences that fail to parse, we can guarantee a high reliability in the understanding and regeneration processes. Figure 3 shows a plot of parse coverage, averaged over weekly intervals, for the first 11 weeks of grammar development.

In addition to monitoring the parsing of weather report data, we periodically evaluate TINA's parse coverage on incoming user queries. This testing is done on a subset of the data, judged to be within domain and not containing any partially uttered words. The evaluation is performed on a pass/fail basis, i.e., we compute the percentage of sentences that achieve a parse in TINA vs. those for which parsing failed. When pooled over several test sets, the average parse coverage is 86% on new, unseen data.

4.3. Understanding

Because the semantic frame is used as input to the JUPITER backend, which is responsible for constructing

```
clause wh_query
   :topic weather
      :quantifier def
         :pred month_date
            :topic date
               :name "tomorrow"
         :pred in
            :topic city
               :name "boston"
```

Figure 4: Example semantic frame for the utterance “What is the weather going to be like tomorrow in Boston?”

the response, it is the logical unit for evaluating understanding. We evaluate the semantic frame at two levels. In the first, we wish to determine what effects proposed changes in the parse rules or the discourse inheritance mechanism may have on the meaning representation. In the second, we evaluate the semantic frame to see how accurately it captured the meaning of the user’s query. For the first, we compare the original frames themselves, with their hierarchical structure intact. Reference frames are stored as text files and read in and converted back to the semantic frame structure by the evaluation program. Evaluation proceeds through the entire semantic frame, recursively evaluating embedded topics and predicates, and flagging cases where the hypothesis frame differs from the reference frame. The evaluation program computes insertions, deletions, and substitutions and finally assigns a score for the match.

Because TINA’s discourse mechanism runs as a separate process within the natural language understanding framework, we can isolate the discourse mechanism from the parser itself. The output of discourse is a new semantic frame with context incorporated. Typically, we create two sets of reference frames for a frozen version of the system, i.e., one with the discourse mechanism enabled and one with it disabled. In order to test only the effects of changes in the parse rules, we use reference frames created with TINA’s discourse mechanism turned off (and disable it when the hypothesis frame is created from the input sentence). Thus, any mismatches between the two frames must be caused by differences in parse rules and the rules that translate from the parser to the semantic frame. In order to test changes to the discourse mechanism, we run the evaluation with reference and hypothesis frames to which discourse has applied.

Even if we understand how changes to the parse rules or discourse mechanism affect the system, we are still left with the most important question for understanding user satisfaction: how well did the system answer the query? To answer this question, we would like to be able to compare the recognizer hypothesis used by the system to answer the subject query with a transcription of the query itself. Furthermore, we want to factor out differences in the recognition string which did not affect understanding (e.g., “a” vs. “the” and even glaring recognition mismatches that are ignored by the natural language component). To make this comparison, we create two new semantic frames from the recognizer hypothesis and the transcription and evaluate them using the comparison program described above.

The frames that are used for evaluating understanding are different from those used for parse and discourse evaluation. As a first step in processing an input semantic frame, JUPITER “flattens” the semantic frame via GENESIS paraphrase rules into a set of keyword-value pairs that are of use to the dialogue component of the JUPITER back-end. This representation is more straightforward to evaluate than the original, more hierarchical, semantic frame, and it contains the critical semantic information needed to answer the query.

Whether evaluating the original semantic frame or its “flattened” counterpart, some differences between semantic frames are not critical for understanding the meaning of an utterance and should be ignored by the evaluation program. We designate certain keys in the frame as “insignificant”, i.e., the values of these keys are not relevant to successful completion of the task. An example of such a key would be “:quantifier” as shown in Figure 4. Our system currently does not make a distinction between questions about the existence of “the thunderstorm in California” and “a thunderstorm in California”. In either case, it will respond with a list of cities in California for which thunderstorms are predicted. We do not want, therefore, to penalize a semantic frame for containing one or the other of these possible values for “:quantifier”. Likewise, certain clause types (e.g., “wh_query” and “identify”) do not have a functional difference for answer generation. To account for this, we allow elements in the semantic frame to be entered into equivalence classes for the purpose of evaluation, much as homophones are used in the calculation of word recognition scores.

4.4. Evaluating the Evaluation Method

Using a test set of 483 utterances from the JUPITER corpus, we compared the output of the automatic evaluation with that of a manual evaluation performed by examining each query-response pair by hand. In judging utterances as “correct”, the automatic evaluation metric agreed with the manual metric 93.3% of the time, i.e., on 291 out of the 312 utterances judged correct. Of the 21 utterances where the two metrics disagreed, 5 were correct purely for pragmatic reasons (e.g., the backend knew to eliminate one of two hypothesized cities that appeared in the semantic frame), 4 were incorrectly judged as correct by the manual metric, and 4 were cases where the *reference* orthography didn’t parse but the hypothesis led to a correct interpretation.

4.5. Logfile evaluation

We have developed a suite of tools to help us examine JUPITER logfiles, created from either simulated or real user interactions. This type of evaluation tells us not only how well the system recognized and understood the user speech, but also how we are doing on database access, i.e., extracting the correct information from the weather report data stored in relational format. We have three separate tools to help us process logfiles. The first enables a system developer to browse through a particular session or a particular day's worth of interactions, listen to the spoken utterance, and see the recognition hypothesis, the *N*-best hypotheses, the transcription, and the system response for each. Although this tool does not evaluate these utterances along any of the dimensions mentioned above, we have found it useful for monitoring system behavior, especially after upgrades have been made. It is also a very interesting, and sometimes humorous way to observe human-machine behavior.

As mentioned above, JUPITER's database of information changes multiple times daily, so that user utterances from any given day cannot be used to recreate a coherent dialogue on a subsequent version of the database. However, we periodically freeze the JUPITER weather database and store this static version as a shadow database. We run a test suite of utterances against this static database and create a logfile of query/response pairs. This test suite is comprised of approximately 5,000 utterances drawn from our pool of spontaneous data, augmented with queries created by system developers to exercise certain aspects of understanding. The new logfile is compared against a reference and changes in system responses are flagged for developers to examine.

Finally, we have a graphical user interface that enables an evaluator to look at individual turns in JUPITER dialogues and categorize them as "correct", "incorrect", "partially correct", or "out-of-domain". This is by far the most time-consuming of our evaluation metrics and one that we hope to replace with the automated understanding evaluation described above.

4.6. Using Understanding to Evaluate Recognition Performance

Once we had the mechanism for evaluating understanding performance, one important way to use it is for determining what effect a new recognizer would have on the understanding component of the system. In the past, we have always assumed that a reduction in error rate would correspond to an increase in understanding and, therefore, user satisfaction. We had no way to quantify this assumption, however, and we were worried that some changes to the recognizer (e.g., adding many new words) might have effects on understanding that might not be reflected in a strict measure of word accuracy. For example, many words that have been added to the recognizer have broad and unclear usage in the linguistic framework, such that parse coverage could be degraded. In March of 1998, we added a significant number of new words to the recognizer, expanding linguistic coverage when feasible to accommodate these words. We generated new *N*-best outputs for a test set of JUPITER utterances and compared these new recognizer hypotheses with the hypotheses generated at the time the data were collected.

System	Word Error	Sentence Error	Understanding Error
Old Recognizer	21.7	42.5	31.7
New Recognizer	16.4	34.4	23.8

Table 3: A comparison of word, sentence, and understanding error rates for two versions of the JUPITER recognizer. This evaluation was done on all utterances, including those with disfluencies, out-of-vocabulary words, and out-of-domain concepts.

Table 3 shows a break-down of system performance between the old recognizer and the new recognizer on a test set of data unseen by both the recognition and understanding components of JUPITER. It should be noted that the error rates are on the entire set, including utterances with significant noise, disfluent speech, and out-of-vocabulary words. We were encouraged to see that we understand approximately 3 out of 4 utterances in this set. Furthermore, the new recognizer contributed to a 25% decrease in understanding error, from 31.7% to 23.8%.

4.7. Evaluating Language Generation

Language generation is a very important component of the JUPITER system. What is actually spoken to the user is a paraphrase of the parsed weather report. We maintain a large file of reference semantic frames from weather report data, along with their associated paraphrases. The outputs of the newest version of the generation component can then be compared against these stored frames and paraphrases. The evaluation component for this part of the system simply flags changes in newly generated paraphrases, and a human evaluator is invoked to judge whether all observed changes are as intended.

5. Summary

Our experience with JUPITER has convinced us that a continuing, reliable source of speech data from users interacting with a real system is an invaluable tool in the development of human language technology. However, in order to exploit these data properly, it is important to develop a set of evaluation metrics and tools we can use to monitor the progress of our systems and identify new areas for research. In the absence of such tools, the data can become overwhelming and we lose sight of how, or if progress is being made.

In complex understanding systems such as JUPITER, evaluating individual components can give us only part of the overall picture. Although it is very important to monitor performance on each individual system module, we must also understand how well JUPITER does in acheiving its ultimate goal, providing informative answers to user queries. The system for evaluating understanding that we describe in this paper is an attempt to address what has, in the past, been a difficult and time-consuming task. As we begin to develop and deploy more spoken language systems, we anticipate a greater need to automate this type of evaluation, especially given the rate at which speech data can be collected.

We continue to develop and refine the evaluation metrics we currently use. In addition, we plan to combine various metrics to give us a better understanding of how recognition, for example, interacts with understanding and how various user characteristics or spontaneous speech phenomena affect each part of the system.

6. Acknowledgements

This research was supported by DARPA under contract N66001-96-C-8529, monitored through Naval Command, Control and Ocean Surveillance Center. We would like to thank Sally Lee for her heroic effort in transcribing the daily onslaught of JUPITER utterances.

7. References

Glass, J., Chang, J., & McCandless, M. (1996). A probabilistic framework for feature-based speech recognition. In *Proc. Fourth International Conference on Spoken Language Processing* (pp. 2277–2280). Philadelphia.

Glass, J., Polifroni, J., & Seneff, S. (1994). Multilingual language generation across multiple domains. In *Proc. International Conference on Spoken Language Processing* (pp. 983–986). Yokohama.

Goddeau, D., Brill, E., Glass, J., Pao, C., Phillips, M., Polifroni, J., Seneff, S. & Zue, V. (1994). GALAXY: a human-language interface to on-line travel information. In *Proc. International Conference on Spoken Language Processing* (pp. 707–710). Yokohama.

Hirschman, L., Bates, M., Dahl, D., Fisher, W., Garofolo, J., Pallett, D., Hunicke-Smith, K., Price, P., Rudnicky, A., & Tzoukermann, E. (1994). Multisite data collection and evaluation in spoken language understanding. In *Proc. DARPA Human Language Technology Workshop* (pp. 19–24). Princeton, NJ.

Hurley, E., Polifroni, J., & Glass, J. (1996). Telephone data collection using the world wide web. In *Proc. Fourth International Conference on Spoken Language Processing* (pp. 1898–1901). Philadelphia.

Price, P., Hirschman, L., Shriberg, E., & Wade, E. (1992) Subject-based evaluation measures for interactive spoken language systems. In *Proc. Fifth DARPA Workshop on Speech and Natural Language* (pp. 34–39). Harriman, NY.

Seneff, S. (1992). TINA: a natural language system for spoken language applications. In *Computational Linguistics*, 18(1), pp. 61–86.

Zue, V., Seneff, S., Glass, J., Hetherington, L., Hurley, E., Meng, H., Pao, C., Polifroni, J., Schloming, R., & Schmid, P. (1997). From interface to content: translingual access and delivery of on-line information. In *Proc. Eurospeech 1997* (pp. 2227–2230), Rhodes.