# SPEECH UNDERSTANDING IN OPEN TASKS

*Wayne Ward, Sunil Issar*
*Xuedong Huang, Hsiao-Wuen Hon, Mei-Yuh Hwang*
*Sheryl Young, Mike Matessa*
*Fu-Hua Liu, Richard Stern*

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

## ABSTRACT

The Air Traffic Information Service task is currently used by DARPA as a common evaluation task for Spoken Language Systems. This task is an example of open type tasks. Subjects are given a task and allowed to interact spontaneously with the system by voice. There is no fixed lexicon or grammar, and subjects are likely to exceed those used by any given system. In order to evaluate system performance on such tasks, a common corpus of training data has been gathered and annotated. An independent test corpus was also created in a similar fashion. This paper explains the techniques used in our system and the performance results on the standard set of tests used to evaluate systems.

## 1. SYSTEM OVERVIEW

Our Spoken Language System uses a speech recognizer which is loosely coupled to a natural language understanding system. The SPHINX-II speech recognition system produces a single best hypothesis for the input. It uses a backed-off class bigram language model in decoding the input. This type of smoothed stochastic language model provides some flexibility when presented with unusual grammatical constructions. The single best hypothesis is passed to the natural language understanding system which uses flexible parsing techniques to cope with novel phrasings and misrecognitions. In addition to the basic speech recognition and natural language understanding modules, we have developed techniques to enhance the performance of each. We have developed an environmental robustness module to minimize the effects of changing environments on the recognition. We have also developed a system to use a knowledge base to asses and correct the parses produced by our natural language parser. We present each of the modules separately and discuss their evaluation results in order to understand how well the techniques perform. The authors on each line in the paper heading reflect those people who worked on each module respectively.

## 2. FLEXIBLE PARSING

Our NL understanding system (Phoenix) is flexible at several levels. It uses a simple frame mechanism to represent task semantics. Frames are associated with the various types of actions that can be taken by the system. Slots in a frame represent the various pieces of information relevant to the action that may be specified by the subject. For example, the most frequently used frame is the one corresponding to a request to display some type of flight information. Slots in the frame specify what information is to be displayed (flights, fares, times, airlines, etc), how it is to be tabulated (a list, a count, etc) and the constraints that are to be used (date ranges, time ranges, price ranges, etc).

The Phoenix system uses recursive transition networks to specify word patterns (sequences of words) which correspond to semantic tokens understood by the system. A subset of tokens are considered as top-level tokens, which means they can be recognized independently of surrounding context. Nets call other nets to produce a semantic parse tree. The top-level tokens appear as slots in frame structures. The frames serve to associate a set of semantic tokens with a function. Information is often represented redundantly in different nets. Some nets represent more complex bindings between tokens, while others represent simple stand-alone values. In our system, slots (pattern specifications) can be at different levels in a hierarchy. Higher level slots can contain the information specified in several lower level slots. These higher level forms allow more specific relations between the lower level slots to be specified. For example, *from denver arriving in dallas after two pm* will have two parses,

[DEPART_LOC] from [depart_loc] [city] denver [ARRIVE_LOC] arriving in [arrive_loc] [city] dallas [DEPART_TIME] [depart_time_range] after [start_time] [time] two pm

and

[DEPART_LOC] from [depart_loc] [city] denver [ARRIVE] arriving in [arrive_loc] [city] dallas [arrive_time_range] after [start_time] [time] two pm

The existence of the higher level slot [ARRIVE] allows this to be resolved. It allows the two lower level nets [arrive_loc] and [arrive_time_range] to be specifically associated. The second parse which has [arrive_loc] and [arrive_time] as subnets of the slot [ARRIVE] is the preferred interpretation. In picking which interpretation is correct, higher level slots are preferred to lower level ones because the associations between concepts is more tightly bound, thus the second (correct) interpretation is picked here. The simple heuristic to select for the interpretation which has fewer slots (with the same number of words accounted for) allows the situation to be resolved correctly.

The parser operates by matching the word patterns for tokens against the input text. A set of possible interpretations are pursued simultaneously. A subsumption algorithm is used to find the longest version of a phrase for efficiency purposes. As tokens (phrases) are recognized, they are added to frames to which they apply. The algorithm is basically a dynamic programming beam search. Many different frames, and several different versions of a frame, are pursued simultaneously. The score for each frame hypothesis is the number of words that it accounts for. At the end of an utterance the parser picks the best scoring frame as the result.

The parse is flexible at the slot level in that it allows slots to be filled independent of order. It is not necessary to represent all different orders in which the slot patterns could occur. Grammatical restarts and repeats are handled by overwriting a slot if the same slot is subsequently recognized again.

The pattern matches are also flexible because of the way the grammars are written. The patterns for a semantic token consist of mandatory words or tokens which are necessary to the meaning of the token and optional elements. The patterns are also written to overgenerate in ways that do not change the semantics. This overgeneration not only makes the pattern matches more flexible but also serves to make the networks smaller. For example, the nets are collapsed at points such that tense, number and case restrictions are not enforced. Articles A and AN are treated identically.

The slots in the best scoring frame are then used to build objects. In this process, all dates, times, names, etc. are mapped into a standard form for the routines that build the database query. The objects represent the information that was extracted from the utterance. There is also a currently active set of objects which represent constraints from previous utterances. The new objects created from the frame are merged with the current set of objects. At this step ellipsis and anaphora are resolved. Resolution of ellipsis and anaphora is relatively simple in this system. The slots in frames are semantic, thus we know the type of object needed for the resolution. For ellipsis, we add the new objects. For anaphora, we simply have to check that an object of that type already exists.

Each frame has an associated function. After the information is extracted and objects built, the frame function is executed. This function takes the action appropriate for the frame. It builds a database query (if appropriate) from objects, sends it to SYBASE (the DataBase Management System we use) and displays output to the user. This system has been described in previous papers. [1] [2]

## 2.1. Natural Language Training Data

The frame structures and patterns for the Recursive Transition Networks were developed by processing transcripts of subjects performing scenarios of the ATIS task. The data were gathered by several sites using Wizard paradigms. This is a paradigm where the subjects are told that they are using a speech recognition system in the task, but an unseen experimenter is actually controlling the responses to the subjects screen. The data were submitted to NIST and released by them. There have been three sets of training data released by NIST: ATIS0, ATIS1 and ATIS2. We used only data from these releases in developing our system. A subset of this data (approximately 5000 utterances) has been annotated with reference answers. We have used only a subset of the ATIS2 data, including all of the annotated data. The development test sets (for ATIS0 and ATIS1) were not included in the training.

## 2.2. Natural Language Processing Results

A set of 980 utterances comprised of 123 sessions from 37 speakers was set aside as a test set. Transcripts of these utterances were processed by the systems to evaluate the performance of the Natural Language Understanding modules. This will provide an upper bound on the performance of the Spoken Language Systems, *i.e.* this represents the performance given perfect recognition. The utterances for sessions provided dialog interaction with a system, not just the processing of isolated utterances. All of the utterances were processed by the systems as dialogs. For result reporting purposes, the utterances were divided into three classes:

- Class A - utterances requiring no context for interpretation
- Class D - utterances that can be interpreted only in the context of previous utterances
- Class X - utterances that for one reason or another were not considered answerable.

Our results for processing the test set transcripts are shown in Table 1. There were 402 utterances in Class A and 285 utterances in Class D for a combined total of 687 utterances. The remainder of the 980 utterances were Class X and thus were not scored. The database output of the system is scored. The percent correct figure is the percent of the utterances for which the system returned the (exactly) correct output from the database. The percent wrong is the percent of the utterances for which the system returned an answer from the database, but the answer was not correct. The percent NO_ANS is the percentage of the utterances that the system did not attempt to answer. The Weighted Error measure is computed as (2 * %Wrong) + %NO_ANSWER. These NL results (both percent correct and weighted error) were the best of any site reporting.

| Class | % Correct | % Wrong | % NO_ANS | Weighted Error |
|---|---|---|---|---|
| A + D | 84.7 | 14.8 | 0.4 | 30.1 |
| A | 88.6 | 11.4 | 0.0 | 22.9 |
| D | 79.3 | 19.6 | 1.1 | 40.4 |

**Table 1:** NL results from processing test set transcripts.

## 2.3. Comparison to February 1991 system

The purpose of evaluations is not only to measure current performance, but also to measure progress over time. A similar evaluation was conducted in February 1991.

**For Class A data**, our percent correct performance increased from 80.7 to 88.6. This means that the percentage of errors decreased from 19.3 to 11.4, representing a **decrease in errors of 41 percent**. The weighted error decreased from 36.0 to 22.9.

**For Class D data**, our percent correct increased from 60.5 to 79.3. The represents a **decrease in errors of 48 percent**. The weighted error was reduced from 115.8 to 40.4.

The basic algorithms used are the same as for previous versions of the system. The increase in performance came primarily from

- Bug fixes (primarily to the SQL generation code)
- Extension of the semantics, grammar and lexicon from processing part of the ATIS2 training data.
- Improved context mechanism

## 2.4. Partial Understanding

In our system, we use the NO_ANSWER response differently than other sites. If our results are compared to others, we output far fewer NO_ANSWER responses. This is because we use a different criteria for choosing not to answer. In order to optimize the weighted error measure, one would want to choose not to answer an utterance if the system believed that the input was not completely understood correctly, *i.e.* if it thought that the answer would not be completely correct. However, if the system chooses not to answer, it should ignore all information in the utterance. Since our goal is to build interactive spoken language understanding systems, we prefer a strategy that shows the user what is understood and engages in a clarification dialog with the user to get missing information or correct misunderstandings. For this procedure we need to retain the information that was understood from the utterance for dialog purposes. The user must also be clearly shown what was understood. Therefore, we only output a NO_ANSWER response when the system did not arrive at even a partial understanding of the utterance.

# 3. SPEECH PROCESSING

For our recognizer, we use the SPHINX-II speech recognition system. In comparison with the SPHINX system, the SPHINX-II system incorporates multiple dynamic features (extended from three codebooks to four), a speaker-normalized front-end, sex-dependent semi-continuous hidden Markov models (which replace discrete models), and the shared-distribution representation (which replaces generalized between-word triphones). [3] [4] For the Feb. 1992 ATIS evaluation, we used SPHINX-II (without the speaker normalization component) to construct vocabulary-independent models and adapted vocabulary-independent models with ATIS training data. The system used a backoff class bigram language model and a Viterbi beam search.

## 3.1. Acoustic Training

In order to efficiently share parameters across word models, the SPHINX-II system uses shared-distribution models. [5] The states in the phonetic HMMs are treated as the basic unit for modeling and are referred to as senones. [4] There were 6500 senones in the systems. Vocabulary-independent acoustic models were trained on approximately 12,000 general English utterances. These models were used to initialize vocabulary specific models (the vocabulary-independent mapping table was used) which were then trained on the task-specific data. Approximately 10,000 utterances from the ATIS0, ATIS1 and ATIS2 training sets were used in the adaptation training. The original vocabulary-independent models were then interpolated with the vocabulary-dependent models to give the adapted models used in the recognition.

## 3.2. Lexicon and Language Model

A backoff class bigram grammar was trained on a total of approximately 12,000 utterances from the same three NIST ATIS distributions. The grammar used a lexicon of 1389 words with 914 word classes defined. The system used seven models for non-speech events.

| Class | Correct | Sub | Deletions | Insertions | Error |
|---|---|---|---|---|---|
| A+D+X | 88.2 | 9.7 | 2.1 | 4.4 | 16.2 |
| A+D | 91.9 | 6.5 | 1.6 | 3.7 | 11.8 |
| A | 92.8 | 5.7 | 1.6 | 3.2 | 10.4 |
| D | 90.3 | 8.2 | 1.5 | 4.8 | 14.5 |
| X | 78.9 | 17.6 | 3.4 | 6.1 | 27.2 |

**Table 2:** SPHINX-II Speech Recognition results.

| Class | % Correct | % Wrong | % NO_ANS | Weighted Error |
|---|---|---|---|---|
| A + D | 66.7 | 32.9 | 0.4 | 66.2 |
| A | 74.1 | 25.9 | 0.0 | 51.7 |
| D | 56.1 | 42.8 | 1.1 | 86.7 |

**Table 3:** SLS results from processing test set speech input.

### 3.3. Speech Processing Results

The Speech recognition results for the test set are shown in Table 2. The Error column is the sum of Substitutions, Insertions and Deletions. The output from the recognizer was then sent to the NL system to get the complete Spoken Language System results. These are shown in Table 3.

### 3.4. Comparison to February 1991 system

**For Class A data**, our word error percentage was reduced from 28.7 to 10.4 representing a **decrease in errors of 64 percent**. The overall SLS error is a function of both the speech recognition and natural language errors. Our percentage of errors in SLS output decreased from 39 to 26 representing a **decrease in errors of 33 percent**. The weighted error decreased from 65.5 to 51.7.

**For Class D data**, our word error percentage was reduced from 26.9 to 14.5 representing a **decrease in errors of 46 percent**. Our percentage of errors in SLS output decreased from 61 to 44 representing a **decrease in errors of 28 percent**. The weighted error decreased from 116 to 87.

The increase in speech recognition performance came from using the SPHINX-II system where we used SPHINX in 1991. The primary differences are:

- Semi-continuous shared-distribution HMMs replaced discrete HMM generalized triphones
- Sex-dependent models were added
- Added second order difference cepstrum codebook

## 4. KNOWLEDGE BASED CORRECTION

The MINDS-II SLS system is a back-end module which applies constraints derived from syntax, semantics, pragmatics, and applicable discourse context and discourse structure to detect and correct erroneous parses, skipped or overlooked information and out of domain requests. MINDS-II transcript processor is composed of a dialog module, an utterance analyzer and a domain constraints model. Input to the CMU MINDS-II NL system is the transcribed string, the parse produced by the PHOENIX caseframe parser and the parse matrix. The system first looks for out of domain requests by looking for otherwise reasonable domain objects and relations among objects not included in this application database. Second, it tries to detect and correct all misparses by searching for alternate interpretations of both strings and relations among identified domain concepts. Further unanswerable queries are detected in this phase, although the system cannot determine whether the queries are unanswerable because the speaker mis-spoke or intentionally requested extra-domain information. Third, the system evaluates all word strings not contained in the parsed representation to assess their potential importance and attempt to account for the information. Unaccounted for information detected includes interjections, regions with inadequate grammatical coverage and regions where the parser does not have the knowledge to include the information in the overall utterance interpretation. All regions containing interjections or on-line edits and corrections are deemed unimportant and passed over. When the system finds utterances with important unaccounted for information, it searches through the parse matrix to find all matches performed in the region. It then applies abductive reasoning and constraint satisfaction techniques to form a new interpretation of the utterance. Semantic and pragmatic knowledge is represented with multi-layered hierarchies of frames. Each knowledge layer contains multiple hierarchies and relations to other layers. Semantic information of similar granularity is represented in a single layer. The knowledge

| System | Class | % Correct | % Wrong | % NO_ANS | Weighted Error |
|---|---|---|---|---|---|
| Phoenix | A + D | 66.7 | 32.9 | 0.4 | 66.2 |
| MINDS-II | A + D | 64.3 | 25.3 | 10.3 | 61.0 |

**Table 4:** UNOFFICIAL Comparison on MINDS-II and Phoenix results from processing test set speech input.

base contains knowledge of objects, attributes, values, actions, events, complex events, plans and goals. Syntactic knowledge is represented as a set of rules. The discourse model makes use of current focus stack, inferred speaker goals and plans, and dialog principles which constrain "what can come next" in a variety of contexts. Goal and plan inference and tracking are performed. Constraints are derived by first applying syntactic constraints, constraining theses by utterance level semantic and pragmatic constraints followed by discourse level constraints when applicable. The system outputs either semantically interpreted utterances represented as variables and bindings for the database interface or error codes for "No_Anwser" items.

The system was trained using 115 dialogs, approximately 1000 of the utterances from the MADCOW ATIS-2 training. Previously, the system had been trained on the ATIS-0 training set. This system incorporates the SOUL utterance analysis system as well as a dialog module for the Feb92 benchmark tests.

### 4.1. Knowledge Based Processing Results

Due to mechanical problems, the results from this test were submitted to NIST after the deadline for official submissions. Therefore, they were not scored by NIST and are not official benchmark results. However, the results were generated observing all procedures for benchmark tests. They were run on the official test set, without looking at the data first. One version control bug was fixed when the system crashed while running the test. No code was changed, we realized that the wrong version (an obsolete one) of one function was used, and we substituted the correct one. The results were scored using the most recent comparator software released by NIST and the official answers (after adjudication).

## 5. ENVIRONMENTAL ROBUSTNESS

This year we incorporated the *Code-Word Dependent Cepstral Normalization* (CDCN) procedure developed by Acero into the ATIS system. For the official ATIS evaluations we used the original version of this algorithm, as described in [6]. (Recent progress on this and similar algorithms for acoustical pre-processing of speech signals are described in elsewhere in these proceedings [7].)

| System | Microphone | % Error |
|---|---|---|
| SPHINX-II | HMD-414 | 13.9 |
| SPHINX-II+CDCN | HMD-414 | 16.6 |
| SPHINX-II+CDCN | PCC-160 | 21.7 |

**Table 5:** Comparison of speech recognition performance of SPHINX-II with and without the CDCN algorithm on the 447 A+D+X sentences in the test set which were recorded using the PCC-160 microphone as well as the Sennheiser HMD-414.

The recognition system used for the robust speech evaluation was identical to that with which the baseline results were obtained except that the CDCN algorithm was used to transform the cepstral coefficients in the test data so that they would most closely approximate the statistics of the ensemble of cepstra observed in the training environment. All incoming speech was processed with the CDCN algorithm, regardless of whether the testing environment was actually the standard Sennheiser close-talking microphone or the desktop Crown PCC-160 microphone, and the algorithm does not have explicit knowledge of the identity of the environment within which it is operating.

Because of time constraints, we did not train the system used for the official robust-speech evaluations as thoroughly as the baseline system was trained. Specifically, the robust-speech system was trained on only 10,000 sentences from the ATIS domain, while the baseline system was trained on an additional 12,000 general English utterances as well. The acoustic models for the robust-speech system using CDCN were created by initializing the HMM training process with the models used in the baseline SPHINX-II system. The official evaluations were performed after only a single iteration through training data that was processed with the CDCN algorithm.

The official speech recognition scores using the CDCN algorithm and the Sennheiser HMD-414 and Crown PCC-160 microphones are summarized in Table 4. We summarize the word error scores for all 447 utterances that were recorded using both the Sennheiser HMD-414 and Crown PCC-160 microphones. For comparison purposes, we include figures for the baseline system on this subset of utterances, as well as figures for the system using the CDCN algorithm for the same sentences. We believe that the degradation in performance from 13.9% to 16.6% for these sentences using the close-talking Sennheiser HMD-414 microphone is at least in part a consequence of the more limited training of the system with the CDCN algorithm. We note that the change from the HMD-414 to the PCC-160 produces only a 30% degradation in error rate. Only two sites submitted data for the present robust speech evaluation, and CMU's percentage degradation in error rate in changing to the new testing environment, as

| System | Microphone | % Correct | % Wrong | % NO_ANS | Weighted Error |
|---|---|---|---|---|---|
| SPHINX-II+CDCN | HMD-414 | 69.0 | 31.0 | 0.0 | 62.0 |
| SPHINX-II+CDCN | PCC-160 | 56.6 | 43.1 | 0.3 | 86.4 |

**Table 6:** Comparison of SLS performance of SPHINX-II with the CDCN algorithm on the 332 A+D sentences in the test set which were recorded using the PCC-160 microphone as well as the Sennheiser HMD-414.

well as the absolute error rate in that environment, were the better of the results from these two sites.

Summary results for the corresponding SLS scores for the 332 Class A+D utterances that were recorded using the Crown PCC-160 microphone are provided in Table 6. Switching the testing environment from the Sennheiser HMD-414 to the Crown PCC-160 degraded the number of correct SQL queries by only 21.8%, which corresponds to a degradation of 39.3% for the weighted error score. CMU was the only site to submit SLS data using the PCC-160 microphone for the official evaluation.

## REFERENCES

1. Ward, W., "The CMU Air Travel Information Service: Understanding Spontaneous Speech", *Proceedings of the DARPA Speech and Natural Language Workshop*, June1990, pp. 127, 129.
2. Ward, W., "Evaluation of the CMU ATIS System", *Proceedings of the DARPA Speech and Natural Language Workshop*, Feb1991, pp. 101, 105.
3. Huang, Lee, Hon, and Hwang,, "Improved Acoustic Modeling for the SPHINX Speech Recognition System", *ICASSP*, 1991, pp. 345-348.
4. Hwang and Huang,, "Subphonetic Modeling with Markov States - Senone", *ICASSP*, 1992.
5. Hwang, M. and Huang X., "Acoustic Classification of Phonetic Hidden Markov Models", *Eurospeech Proceedings*, 1991.
6. Acero, A. and Stern, R. M., "Environmental Robustness in Automatic Speech Recognition", *ICASSP-90*, April 1990, pp. 849-852.
7. Stern, R. M., Liu, F.-H., Ohshima, Y., Sullivan, T. M., and Acero, A., "Multiple Approaches to Robust Speech Recognition", *DARPA Speech and Natural Language Workshop*, February 1992.