The CMU Air Travel Information Service: Understanding Spontaneous Speech

Wayne Ward
School of Computer Science
Carnegie Mellon University
Pittsburgh, Pa 15213

Introduction

Understanding spontaneous speech presents several problems not found in processing read speech input. Spontaneous speech is often not fluent. It contains stutters, filled pauses, restarts, repeats, interjections, etc. Casual users do not know the lexicon and grammar used by the system. It is therefore very difficult for a speech understanding system to achieve good coverage of the lexicon and grammar that subjects might use.

The Air Travel Information Service task is being used to develop and evaluate speech understanding systems for database query like tasks. In the ATIS task, novice users are asked to perform a task that requires getting information from the Air Travel database. This database contains information about flights and their fares, airports, aircraft, etc. Users compose the questions themselves, and are allowed to phrase the queries any way they choose. No explicit grammar or lexicon is given to the subject.

At CMU, we are developing a system, called Phoenix, to understand spontaneous speech. We have implemented an initial version of this system for the ATIS task. This paper presents the design of the Phoenix system and its current status. We also report results for the first ATIS evaluation set distributed by NIST.

The Phoenix System

The problems posed by spontaneous speech can be divided into four categories

- User noise - breath noise, filled pauses and other user generated noise
- Environment noise - door slams, phone rings, etc.
- Out-of-vocabulary words - The subject says words that the system doesn't know.
- Grammatical coverage - Subjects often use grammatically ill-formed utterances and restart and repeat phrases.

Phoenix address these problems by using non-verbal sound models, an out-of-vocabulary word model and flexible parsing.

Non-verbal sound models

Models for sounds other than speech have been shown to significantly increase performance of HMM-based recognizers for noisy input. [5] [7] In this technique, additional models are added to the system that represent non-verbal sounds, just as word models represent verbal sounds. These models are trained exactly as if they were word models, but using the noisy input. Thus, sounds that are not words are allowed to map onto tokens that are also not words.

Out-of-vocabulary word model

In order to deal with out-of-vocabulary words, we are using a technique essentially the same as the one presented by BBN. [1] We have an explicit model for out-of-vocabulary words. This model allows any triphone (context dependent phone) to follow any other triphone (given of course that the context is the same) with a bigram probability model. The bigrams are trained from a large dictionary of English pronunciations.

Flexible parsing

We use a frame based parser similar to the DYPAR parser used by Carbonell, et al. to process ill-formed text, [2] and the MINDS system previously developed at CMU. [8] Semantic information is represented by a set of frames. Each frame contains a set of slots representing pieces of information. In order to fill in the frames, we use a partitioned semantic phrase grammar. The grammar is a semantic grammar, non-terminals are semantic concepts instead of parts of speech. The grammar is also written so that phrases can stand alone (be recognized by a net) as well as being embedded in a sentence. Strings of phrases which do not form a grammatical English sentence are still parsed by the system. The grammar is compiled into a set of finite-state networks. Networks can "call" other networks, thereby significantly reducing the overall size of the system. These networks are used to perform pattern matches against word strings. The grammar is partitioned, instead of one big network, there are many small networks. Each slot type is represented by a separate network which specifies all ways of saying the meaning represented by the slot. This general approach has been described in an earlier paper. [6]

The operation of the parser can be viewed as "phrase spotting". A beam of possible interpretations are pursued

Source	Number True	Number False	No Answer	Percent Correct
Transcript	45	47	1	48
Speech	36	57	0	39

Table 1: Results as scored by NIST.

Source	Number True	Number False	No Answer	Percent Correct
Transcript	60	32	1	65
Speech	39	54	0	42

Table 2: Rescored results

simultaneously. An interpretation is a frame with some of its slots filled. The finite-state networks perform pattern matches against the input string. When a phrase is recognized, it attemps to extend all current interpretations. This amounts to dynamic programming on series of phrases. The score for an interpretation is the number of input words that it accounts for. At the end of the utterance, the best scoring interpretation is output.

System Structure

The overall structure of the system is shown in Figure 1. We use the Sphinx system as our recognizer module. [4]. Currently, it is a Top-1 system. That is, the recognizer and parser are not integrated. The grammar used by the parser is used to generate a word pair grammar. The recognizer uses the word pair grammar in decoding the speech input. The recognizer produces a single best hypothesis. This hypothesis is then passed to the frame-based parser which assigns word strings to slots in a frame as explained above.

The slots in the frame are then mapped to canonical form. This puts all dates, times, names, etc. in a standard form for the routines that build the database query. At this step ellipsis and anaphora are resolved using current objects built as a result of previous utterances. Objects consist of currently active constraints, the set of flights that meet the constraints and a list of individual flights in focus. Resolution of ellipsis and anaphora is relatively simple in this system. We are aided greatly by the fact that the slots in frames are semantic, thus we know the type of object needed for the resolution. The canonical frame represents the information that was extracted from the utterance. It is then used to build a database query. This query is sent to the SYBASE database management system and the returned results are displayed to the user.

Results

Our current system has a 484 word vocabulary and a word pair grammar with perplexity 85. We use the Vocabulary-independent phone models generated by Hon. [3] We have not yet added the non-verbal and out-of-vocabulary models to the system. The only technique currently used to cope with spontaneous speech is a word pair grammar and a flexible parser.

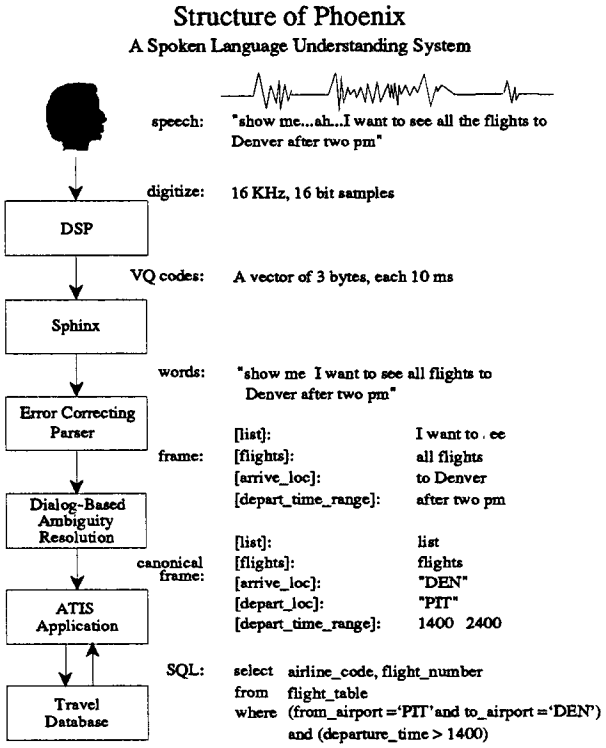

Figure 1: Structure of the Phoenix system

The test data consists of a total of 93 utterances taken from five speakers. The data was gathered by TI and distributed by NIST. All utterances were "Class-A". Both transcript and speech input were processed. The database output in CAS format was sent to NIST where it was scored against the reference database answers. Table 1 shows the results of the NIST evaluation.

As a result of errors in generating the output to be scored, a significant number of utterances that parsed correctly were scored as incorrect. Most of these were of three types that resulted from a misunderstanding on my part as to what was to be generated.

- Dates - Our system generated dates relative to the time the system was run instead of relative to when the corpus was gathered.

Sourc	Subs	Del	Ins	Error
Word	25	15	4	44
String	97	-	-	97

Table 3: Recognition error rates (percentages)

- Abbrevations - We printed codes or abbreviations rather than the full text description as an answer.
- Round-trip - The test for round-trip fares (of flights) was incorrectly applied.

Output in these situations was correct given the (incorrect) assumptions that I used. In order to understand the system's behavior, it is useful to look at the scores if the three bugs were fixed. This more fully reflects the true abilities of the system. After sending our output to NIST, I fixed these three bugs (total time under three hours) and reprocessed the test data. Table 2 presents the same test data after these changes.

Analysis of trace ouput for the data showed that 75 percent of the transcript utterances parsed correctly. The additional degradation to 65 percent is a result of other errors in generating database queries.

It is also interesting to examine the word and string error rates for the recognizer output. These are shown in Table 3. A string error rate of 97 percent means that only three percent of the utterances contained no errors. However, 42 percent of the utterances gave correct answers. This illustrates the ability of the parser to handle minor misrecognitions in the recognized string. The word error rate of 44 percent is poor given the high quality of the basic recognizer and relatively low perplexity of the word pair grammar. We feel that this will improve considerably with the addition of non-verbal and out-of-vocabulary models and with better lexical and grammatical coverage.

References

1. Asadi, A., Schwartz, R., Makhoul, J. Automatic Detection Of New Words In A Large Vocabulary Continuous Speech Recognition System. Proceedings of the DARPA Speech and Natural Language Workshop, 1989, pp. 263, 265.

2. Carbonell, J.G. and Hayes, P.J. Recovery Strategies for Parsing Extragrammatical Language. Tech. Rept. CMU-CS-84-107, Carnegie-Mellon University Computer Science Technical Report, 1984.

3. Hon, H.W., Lee, K.F., Weide, R. Towards Speech Recognition Without Vocabulary-Specific Training. Proceedings of the DARPA Speech and Natural Language Workshop, 1989, pp. 271, 275.

4. Lee, K.-F. *Automatic Speech Recognition: The Development of the SPHINX System.* Kluwer Academic Publishers, Boston, 1989.

5. Ward, W. Modelling Non-verbal Sounds for Speech Recognition. Proceedings of the DARPA Speech and Natural Language Workshop, 1989, pp. 47, 50.

6. Ward, W. Understanding Spontaneous Speech. Proceedings of the DARPA Speech and Natural Language Workshop, 1989, pp. 137, 141.

7. Wilpon, J.G., Rabiner, L.R., Lee, C.H., Goldman, E.R. Automatic Recognition of Vocabulary Word Sets in Unconstrained Speech Using Hidden Markov Models. in press Transactions ASSP, 1990.

8. Young, S. R., Hauptmann, A. G., Ward, W. H., Smith, E. T. and Werner, P. "High Level Knowledge Sources in Usable Speech Recognition Systems". *Communications of the ACM 32*, 2 (1989), 183-194.