UNDERSTANDING SPONTANEOUS SPEECH

Wayne Ward[1]
Carnegie Mellon University
Computer Science Department
Pittsburgh, PA 15213

ABSTRACT

When speech understanding systems are used in real applications, they will have to deal with phenomena peculiar to spontaneous speech. People use language differently when they speak than when they write. Spoken language contains many interjections, filled pauses, etc. Speakers often don't use well-formed sentences. They speak in phrases, have restarts, etc. Systems designed for written or read text will encounter serious difficulties processing such input. This paper outlines our strategy for dealing with spontaneous spoken input in a speech recognition system.

INTRODUCTION

As systems become more habitable and allow users to speak naturally, speech recognizers and parsers are going to have to deal with events not present in written text or read speech. Spontaneous speech contains a number of phenomena that cause problems for current systems.

- filled pauses - noises made by the speaker that don't correspond to words (ah, uh, um, etc).
- restarts - repeating a word or phrase. The original word or phrase may be complete or truncated.
- interjections - extraneous phrases as in "on line thirty, I guess it is".
- unknown or mispronounced words
- ellipsis
- ungrammatical constructions - Users make errors of agreement (sub-verb, number, etc) and may use constituents in unusual orders ("to the utilities cell add fifty dollars").

These phenomena violate constraints currently used by speech recognizers to increase performance. This can cause complete recognition failure for an utterance.

In his paper on habitability, Watt (1968) characterizes the problem as a difference between **COMPETENCE** and **PERFORMANCE**. We must recognize what people say, not what they think is grammatical. In real dialogs, much can be understood from context and is left out of utterances. Ellipsis is very common. Many elliptical utterances are not just deletions from expected well-formed sentences. Consider the utterances "okay .. expenses .. mortgage seven forty eight point fifty seven .. car payment . two forty three . point twenty seven . bank surcharge . fifteen dollars". The focus is the information to be transferred, a label specification and an amount. Each utterance is the simplest expression of the neccessary information with no other embroidery.

The solution to this problem must involve both parsing and recognition strategies. It must resolve the competeing aims of reducing search space and remaining flexible to the unexpected. Our approach is a combination of specific modelling of acoustic properties and a flexible control structure.

---

[1]This research was sponsored by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 5167, under contract number N00039-85-C-0163. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

LIMITATIONS OF CURRENT RECOGNIZERS

Current state-of-the-art speech recognition systems make several assumptions about the input in order to increase performance:

- A single well-formed sentence is spoken. Well-formed means acceptable to the system's language model.
- Only words in the system's lexicon are used.
- The sentence is delimited by pauses with no internal pauses.
- There is no extraneous noise. Every part of the input should be matched against a word model.

These assumptions allow the system to enforce constraints of continuity and gramaticallity. That is, they attempt to find a grammatical sequence of words that spans the entire utterance. Some word model (or silence) must be matched against all areas of the input. The input is searched left-to-right for legal sequences of words. Previously recognized word boundaries are used as the starting point for subsequent words and only words constituting legal extensions of current paths are considered. Legal word sequences are defined by a language model. This model may be a grammar or sequence transition probabilities derived from a corpus. If the recognizer does not correctly recognize a portion of the input, for subsequent portions of the input it is no longer searching for the correct words at the correct boundaries. This leads to misrecognition, and the user has no option but to repeat the sentence, perhaps rephrasing it.

These constraints serve to reduce the search space for an utterance. Giving up grammar constraints during recognition may allow the system to recover more quickly after an error, but there will be more errors in well-formed utterances due to lesser constraint and the resulting strings must still be parsed. Likewise, word-spotting (starting every word at every frame) to produce a word lattice is not enough. Words must still be joined into sequences to form a sentence. It is neccessary to allow interruptions in the grammar and in the recognition. The recognizer must be allowed to search for words that do not form grammatical extensions of a current hypothesis. It must also allow some areas to go unmatched (in the case of an unknown word).

TECHNIQUES FOR TEXT INPUT

Many of the same types of problems exist in typed natural language interfaces. Work has previously been done on parsing typed extra-grammatical input of this sort (Carbonell & Hayes 1984, Hayes & Carbonell 1981, Weischedel & Black 1980, Weischedel & Sondheimer 1987). Hindle (1983) processed transcripts of speech using a Mracus-style parser. This work basically represents two approaches to handling ill-formed input:

1. Look for patterns in the syntax and have an associated action for each pattern. These methods require finding the "editing signal" which indicates a specific pattern that the system knows how to recover from.
2. Look for gaps or redundancies in the semantics. Account for as much of the input as possible and then use the overall semantics to help define the proper response.

Carbonell & Hayes (1984) point out the importance of semantic information in parsing extra-grammatical input. The notion is to "step back", that is look at the other portions of the utterance and look for gaps or repetitions in semantic information. They discuss the suitability of three general parsing strategies for recovering from ill-formed input and ellipsis.

- Network Parsers - These include ATN's and semantic grammars. It is very hard to "step back and take a broad view" with these parsers. Too much is encoded locally in state information. Networks are naturally top-down left-to-right oriented.
- Pattern Matching Parsers - Partial pattern matches can be allowed which gives some ability to "step back",abut there is no natural way to differentiate between how important constituents are. That is, the grammar is "uniformly represented".
- Case Frame Parsers - These allow the ability to "step back". They provide a convienient mechanism for using semantic and pragmatic information. Semantic components or cases can be compared instead of syntactic structures. "In brief, the encoding of domain semantics and canonical structure for multiple surface manifestations makes case frame instantiation a much better basis for robust resolution than semantic grammars."

The general idea is to isolate the error and use recognized areas on both sides to give more information as to what is missing or repeated. The entire utterance is parsed, filling in as much of the case frame as possible. If there is unparsed input and the frame is complete, the input can be treated as spurious. If there is a gap in the structure (unfilled elements) then the unrecognized element was probably a filler for that component. If the same case is filled by more than one element, then the first can be ignored. The user should be made aware of any of these conditions. If there is a gap in the semantics, the system must engage in a clarification dialog with the user. This interaction can be very focused since the system now has an expectation of the semantic type that is missing. Unfortunately, we cannot use their recovery strategies *directly*. We wish to use grammar predictively to constrain the word search. In speech the correct input string is not known and only strings that are searched for are produced. For example, it is obvious in a typed interface when the system is given an unknown word. A speech recognizer will never produce a word not in its lexicon. The effect of an unknown word in the input is that all words in the system lexicon that are legal extensions of current paths are matched against that area of the input. Those that match sufficiently well will extend their paths across the area, but the correct word will of course not be searched for. Unless some other word has an acceptable acoustic match and similar grammatical role, no path will be correctly aligned with the input. Similarly, such a system will never produce a restart sequence unless it is specifically searched for. As in the text input systems, we wish to use sentence fragments on both sides of a problem area to help determine what is missing. This means being able to recognize portions of the utterance that follow an unrecognized region. For this we must depart from the strict left-to-right grammatical extension control strategy.

PROCESSING SPONTANEOUS SPEECH

At CMU we are developing a system (called Phoenix) for recognizing spontaneous speech. This system uses the HMM word models developed in the Sphinx system (Lee 1989). It relies on specific modelling of acoustic features and a flexible control structure to process natural speech. We are currently implementing this system for a spreadsheet task.

We want to specifically model the acoustic features of spontaneous speech. This includes phenomena like lengthening phonemes and filled pauses. We created new phonemes and words for several classes of filled pauses(uh, er, um, ah, etc). We are gathering a corpus of spontaneous speech for users engaged in a spreadsheet task. The phone models for the system will be trained on this corpus. This training will be in addition to, not instead of the current training set.

The control structure for the recognizer is based on recognizing phrases rather than sentences. Input is viewed as a series of phrases instead of sentences with well defined boundaries. The system has a grammar which defines legal word sequences. These represent complete sentences as well as phrases which aren't embedded in a sentence. A phrase may be as short as a word or as long as a complete sentence. The system has a set of "meanings" or concepts which represent the information to be transferred. Each meaning is represented by a network that contains all surface strings or phrases for expressing the concept. Additionally there are semantic structures which represent the actions that the system can take. These structures are very similar to case frames in that they contain slots for meanings or information required to complete an action. Unusual constituent ordering is allowed by allowing meanings within a structure to occur in any order.

The input is processed left-to-right using the grammar to search for phrases. All phrases are searched for after detection of a pause or interruption. Phrases are not deleted when they can no longer be extended. As phrases are recognized, they are assigned a meaning and attached to the appropriate semantic structures. A single phrase or sequence of phrases may be necessary to complete the semantics of a structure. No single structure may contain phrases overlapping in time and multiple structures may be competing for instantiation.

The idea is to concentrate on recognition of "meaning units" not sentences. Phrases themselves must be well-formed but need not combine into a grammatical sentence. Grammar is used as a local constraint to govern the grouping of words into phrases. Global constraints come from the semantics of the system which govern the combining of a sequence of meanings into a defined action.

With this system we can process spoken input with strategies similar to those used by Carbonell & Hayes. Here there is a set of possible paths being evaluated rather than a single one. The various phenomena can now be characterized by the semantics of the entire utterance.

- Missing or unknown words - There will not be an unknown word in the recognized string. There will be

either an incorrect word or an unmatched area. These words may be important, that is represent semantics necessary for interpreting the utterance, or they may be extraneous. If they are extraneous, the frame will be complete and they may be ignored. If they are important, there will be a gap in the semantics. A slot will be unfilled in an otherwise complete frame.

- Spurious words or phrases - These will leave part of the input unaccounted for but the utterance will be semantically complete.
- Restarts - The restarted phrase may be truncated or complete. If complete, the structure will have two phrases competing for the same slot. In this case, the first phrase can be ignored. In the case of a truncated phrase, the structure will have a gap in its coverage of the input but the semantics will be complete. In this case the truncated phrase is ignored. Truncated phrases are an explicit signal to look for a restart.
- Out of order constituents - are not a problem since no ordering is imposed.
- Elliptical or telegraphic input - The system naturally recognizes these. They represent speaking only the neccessary information with minimal phrasing. Semantic structures provide a convienient mechanism for specifying what is "understood" in a situation and therefore can be left out of the utterance.

As an example, consider processing a restarted phrase like "go down a screen .. screen's worth". This is an example of a PAGE command with the slots [move-up] [integer] [screen]. The individual phrases are recognized as

([move-up] go down)
([integer] a)
([screen] screen)
([screen] screen's worth).

Phrases on both sides of the discontinuity are recognized and used to complete a structure. The second instance of the [screen] meaning superseedes the first giving the correct interpretation "go down a screen's worth".

It is not sufficient to simply ignore unrecognized areas without classifying them. Consider the sequence "under finance enter fifty dollars ... under utilities enter thirty dollars .. under credit card enter ten dollars". If "finance" is not in the lexicon (and therefore not recognized), the system can't simply ignore it and go on. This would result in the erroneous parse "enter fifty dollars under utilities". This sort of problem is less severe in an interactive situation than when processing in the background. Prosodic cues can be very useful in resolving this type of situation. Initially we are filtering out filled pauses, interjections and cue phrases. The only prosodic features used are pauses. Later we will incorporate these into the system since they are useful in resolving ambiguous situations. In the last example, if the input had been "under finance enter fifty dollars .. okay.. under utilities enter thirty dollars .. fine, now under credit card enter ten dollars", the cue phrases "okay" and "fine now" would indicate that "enter fifty dollars" associated with some unrecognized item ("finance") while "enter thirty dollars" associates with "utilities".

Recovery cannot always be automatic. It will sometimes be neccessary to interact with the user to resolve the problem. However, since the system has information as to what is most likely missing (the unfilled slots) the interaction can be much more focused than a general request to repeat or paraphrase.

In order to deal with unknown or mispronounced words, we must have better estimates of the quality of a recognized string. Currently most recognizers represent a path by a single score which represents its overall quality. There is no indication of whether some parts of the input are very good matches and others very poor or the quality was fairly uniform. The quality of the acoustic match can be monitored at several levels (vq, state, phoneme, word, phrase, structure) and the resulting pattern used to help classify the recognition. Quality is a relative term here. We propose to keep running means and variances for the speaker at each of these levels so that variances from the norm for this speaker not absolute measures will be used. This will aid the system in detecting when a correct path is going awry. The system will of course not produce an unknown word but it can detect that no acceptable matches are found for a region.

SUMMARY

We aim to achive robust recognition by using a mixed strategy of syntax and semantics. Grammar is used locally to form phrases from words. The phrases are associated with meanings and semantic constraints are applied to sequences of meanings. This allows us to use grammar to guide the word search without insisting that the final results conform to the grammar. The focus is on the information to be transferred, phrases convey meanings.

Sequences of meanings more naturally represent performance, particularly ellipsis and telegraphic style, than other mechanisms in use. Using semantics from all recognized parts of an utterance helps resolve ambiguous or ill-formed sections.

References

1. 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.

2. Hayes, P.J. and Carbonell, J.G. Multi-Strategy Parsing and Its Role in Robust Man-Machine Communication. Tech. Rept. CMU-CS-81-118, Carnegie-Mellon University Computer Science Technical Report, 1981.

3. Hindle, D. Deterministic Parsing of Syntactic Non-fluencies. ACL83, 1983, pp. 123 - 128.

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

5. Watt, W. C. Habitability. American Documentation, 1968, pp. 338-351.

6. Weischedel, R.M. and Black, J.E. "Responding Intelligently to Unparsable Inputs". *American Journal of Computation Linguistics 6* (1980), 97-109.

7. Weischedel, R.M. and Sondheimer, N.K. Meta-rules as a Basis for Processing Ill-formed Input. In *Communication Failure in Dialogue and Discourse*, Reilly, R.G., Ed., North-Holland, 1987.