3rd European Conference on
Speech Communication and Technology
EUROSPEECH'93
Berlin, Germany, September 19-23, 1993

SEMANTIC AND PRAGMATICALLY BASED RE-RECOGNITION OF SPONTANEOUS SPEECH

Sheryl R. Young and Wayne Ward

School of Computer Science, Carnegie Mellon University
Pittsburgh, Pennsylvania 15213 USA

10.21437/Eurospeech.1993-504

Abstract

This paper describes a novel architecture and algorithms for combining stochastic modeling and Natural Language Understanding techniques to help speech recognition and understanding. In this system, an utterance is initially processed by a speech recognizer using a standard class bigram language model to produce a single best scoring word string. This word string is then parsed by the Phoenix parser [1], which produces a semantic frame. The parser uses Recursive Transition Networks to represent semantic fragments, or word strings which are meaningful to the system. Semantic fragments of the utterance are assigned to slots in frames. Semantic, pragmatic and discourse knowledge is then applied to the parsed frame to identify misrecognized substrings and develop content predictions for the misrecognized regions. For this, we compute within utterance semantic constraints, constraints arising from speech repair acts (e.g. on-line edits and corrections) as well as dialog-based constraints arising from different types of sub-dialogs (or what have traditionally been called discourse and domain plans) and the content of prior inputs and system responses. The predictions correspond to a small subset of the semantic networks known to the system. The region boundaries of the input along with the set of predicted semantic networks are passed to a Recursive Transition Network speech decoder which uses them in re-recognizing the specified region of the utterance. The networks used by the RTN decoder are the same ones used by the parser. Only the predicted subset of nets are used in the re-recognition. We describe our algorithms for detecting misrecognitions and generating predictions as well as the operation of our RTN-based recognizer. The system was trained on training data from the ARPA Air Travel Information Service (ATIS) task, and tested on an independent test set of 1000 utterances.

1. Introduction

Spontaneous speech is both acoustically and grammatically challenging to recognize. Acoustically, you encounter filled and silent pauses, human and environmental noise, stutterring and truncated or partial words. Grammatically, spontaneous speech contains mid-utterance corrections and verbal edits [2, 3], out-of-vocabulary words, meta level comments, dysfluencies, ungrammatical constructions and partial utterances. While traditional grammar based (Finite State, LR, Recursive Transition Network) recognizers have been used successfully for decoding read speech, the standard implementations are less successful for spontaneous speech. They are less robust than stochastic language model recognizers to the disfluent, ungrammatical and verbally "corrected" utterances encountered in spontaneous speech. These difficulties are primarily caused by the challenge of generating grammatical rules that cover commonly occurring spontaneous phenomena. It is very difficult to generate rules that provide good coverage of the word sequences people produce when speaking spontaneously.

Stochastic language models are more robust to unseen word sequences since they can be smoothed and their scope is short enough to allow search realignment after an error. However, stochastic language models may provide a poor match with the Natural Language understanding portions of the system. They do not enforce applicable syntactic, semantic, pragmatic and dialog constraints, and often produce word strings that can't be parsed by standard NL parsers. Some success in understanding spontaneous speech has come from using stochastic language models to decode an utterance and then using a flexible Natural Language parser to process the decoded string [4, 3, 1]. This is a loosely coupled system, in that the recognizer and parser use different language models. While this provides robust decoding, it does not take advantage of the longer span constraints provided by the rule based grammars, and misrecognitions may still cause parser errors. The system described and evaluated in this paper is designed to cope with spontaneous speech phenomena. It uses both stochastic and rule-based knowledge and attempts to achieve compatibility between the recognition and understanding functions of the system while remaining robust to unexpected input.

2. System Architecture

The system uses a two-pass architecture. On the first pass a word-class bigram based decoder is used to produce the single best word string hypothesis. This pass is both efficient and robust, but is likely to have minor errors. The output word string is parsed by the Phoenix parser into slots in a semantic frame. Portions of the input which can't be accounted for by the parser are left out of the parse. The parser output and the hypothesized word string are then analyzed to determine possible misrecognitions by the Minds-II system.

This system keeps track of all preceeding interaction (user input, system responses and changes in screen display). When a string of words is flagged as being potentially misrecognized, Minds-II generates content predictions for the potentially misrecognized substring. These predictions are derived from the current applicable context, inferred speaker goals and plans, the dialog history, sentence-level semantic constraints and applicable discourse plans. The predictions specify the most likely concepts that should have been in the region of speech corresponding to the word string. These concepts correspond to a set of slot networks used by the parser. The system can also generate predictions to restrict the expansion of specified networks. These predictions would generally arise from dialog context.

For example, consider the mis-recognition of flight numbers in an airline travel task, such as substituting *U S 150* for *U S 115*. Both of the strings are legitimate flight numbers and therefore semantically and pragmatically consistent. However, if the utterance happened to be in context where the user was looking at a list of flights containing *U S 115* but not *U S 150*, we have a mechanism to restrict the network *flight_number* to only those flights in the current list.

As illustrated in Figure 1, prediction based net restriction is done by taking advantage of the RTN representation. In our compiled networks, a call arc (one that represents a call to another network) simply contains the number of the network being called. We have an array which contains the start address in memory for each network, and the address to jump to is found by indexing into this array. In order to restrict the expansion of a given net, we dynamically compile a version which contains only the desired strings and use it for re-recognizing the potentially misrecognized word string. The address of the new net then temporarily replaces the original address for the network in the index array, so that whenever the net is called, the new version is used. When the constraints on the net are dropped, the old address is written back into its index in the array and the space for the new net is freed.

In the second pass, the predictions and starting and ending times for the region are passed to a second recognizer. This recognizer uses Recursive Transition Networks for its language model, and will only produce sequences of words allowed by the networks. Only the predicted networks are used in the decoding, so this is a low perplexity search that will produce a string of words acceptable to

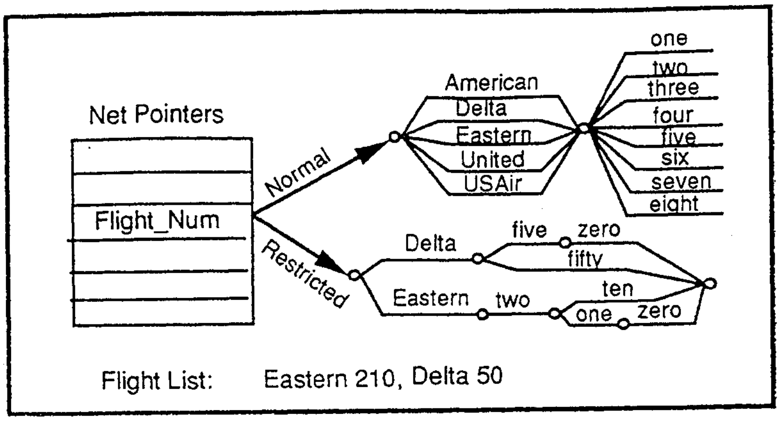

Figure 1: Dynamically Derived Net Restrictions

the parser or nothing at all. The concept networks are matched only against the specified region of the utterance. The output from the rerecognition is then used to replace the original misrecognized string and inserted into the parsed frame.

3. Identifying Misrecognized Strings

The system uses a number of techniques to analyze the current word string hypothesis and parse to try to detect errors. These can be parse errors [5] or recognition errors. Basically, the system relies upon domain semantics, abductive reasoning, constraints derived from prior discourse including contextually appropriate subdialogs and topic changes as well as constraints upon current topics of discussion and objects and attributes available for reference. To do this, the system employs three data structures, a frame-based knowledge base, a domain plan tree for inferring and tracking speaker goals, plans and plan failures [6], and a focus stack. The domain tree and focus stack are also used by the discourse model for tracking discourse plans [7]. The discourse model determines the types of subdialogs that can be initiated at a given point in the interaction and computes contextual constraints upon their content [8]. The system uses it semantic, pragmatic and discourse knowledge to correct parse errors and semantically or pragmatically inconsistent recognition errors. It cannot detect semantically consistent misrecognitions such as:

I would like information on ONE WAY flights from Boston to Pitt

I would like information on MONDAY flights from Boston to Pitt

The system generates a set of constraints that the error-ful utterance must fulfill. In some cases, the constraints are restrictive enough to permit the system to correct an error without rerecognizing any portions of the utterance. This usually occurs when an object is misrecognized and substituted for a similar object that is not available for reference. This is common with flight number misrecognitions or clarifications of objects / attributes presented in the last database answer. If the system substitutes a not-available-for-reference item and there is only a single, similar item that is contextually appropriate, the system can correct the substitution error. For the rest, predictions are generated to guide the rerecognition.

Rerecognition predictions are used to guide the second-pass recognition of substrings of the original recognized output utterance. Not only parsed words can be sent for rerecognition. Words in the word string hypothesis which are left out of the parse are candidate for rerecognition, as long as they aren't isolated short function words.

In order to perform a correction, either with our without rerecognition, the system looks for semantic and pragmatic inconsistency at both the utterance and discourse level. Within an utterance, each of the phrases and words must modify or complement one another. Across utterances, contextual constraints on what is referencible, what subjects are available for discussion, and what types of subdialogs can be initiated must be adhered to. Plan tree traversal heuristics [9] that indicate what topics are available for discussion and requisite ordering among topics or plan steps must not be violated. For example, a subtree or plan step must be completed before another is begun and completed plan steps should not be repeated in the absence of a plan failure or a correction subdialog. Semantic and pragmatic consistency is computed primarily by applying abductive reasoning and constraint satisfaction techniques. Abductive reasoning is used to evaluate consistency within the hypothesis. It decides which phrases modify one another and how they can be combined to form one or more meaningful utterances. Essentially, the process uses the representations of identified knowledge base entries and determines the ways in which they can be related to one another, while maintaining consistency among the entire set of entries. This knowledge is used in conjunction with constraint satisfaction to determine referencible objects and to compute those objects actions and attributes that are currently in focus or contextually appropriate. To determine which portion of an utterance (if any) is most likely to be inaccurately recognized, the analysis routines try to build the best, most encompassing semantically and pragmatically consistent representation of an utterance. It takes into account heuristics for processing restarts and mid-utterance corrections and tries to build a single semantic representation of an utterance, identifying the least number of semantic objects or attributes that are inconsistent.

The conceptual analyzer not only determines what possible phrases compose a meaningful request or statement, it also identifies combinations of meanings which violate domain constraints. These constraint violations include both type constraints of objects and attributes and n-tuple constraint violations. In helping to identify mis-recognitions, these constraints assist in determining whether portions of a hypothesis can reasonably modify one another. To illustrate the use of constraints, consider the following rule for long range transportation taken from the current knowledge base:

```
Long Range Transportation
```

Objects:
vehicle *long-range vehicle,*
origin *location,*
destination *location*
objects-transported *object*
Relations:
origin - destination

Here we have constraints on the type of objects that may fill these roles and relational constraints. In this example, relational or tuple constraints put restrictions on the relationship between the origin and destination slot fillers.

In the context of a dialog, we use the state of the world, discourse structure constraints, inferred current goals and plans, and a current focus stack. We detect inconsistencies between the current parse and these global structures.

4. Generating Predictions

The mechanisms used for predicting concepts in a misrecognized region as similar to those used for detecting the misrecognition. Those that rely only on information within the utterance are constraint satisfaction in conjunction with abductive reasoning and basic syntactic knowledge of constituents and attachment. These are general, domain independent techniques which rely upon a domain specific knowledge base. The system begins by hypothesizing which entities and actions in the remainder of the hypothesis the identified region could modify. This process relies primarily on syntactic knowledge. Next, it uses the domain knowledge base and constraint satisfaction techniques to hypothesize reasonable semantic values that the region could take. To determine the meaning of an unknown region of input, the reasoning component searches the knowledge base to determine the most general, contextually appropriate concepts that are consistent with the other concepts in the utterance.

Conceptual analysis also helps determine possible meanings for a misrecognized phrase by eliminating and bounding possible hypotheses. The system generates a set of hypothesized phrase meanings for each portion of an utterance marked as misrecognized.

When the utterance is in the context of a dialog, the hypothesis set can be further constrained. The additional constraints arise by computing consistency with the state of the world as represented in the dialog model. The stack of goals and plans, along with domain independent properties of dialog structure are used to compute consistency and to determine whether any portion of an utterance is highly improbable.

Here, the system begins by identifying the discourse action (continue plan, clarify last database response or last utterance, confirm contents of plan subtree, etc.) likely to be executed based upon the "undisputed" information in the utterance. This step rarely results in any ambiguity ragarding discourse plans, although occasionally multiple hypotheses for continuing the current domain plan step must be maintained. Once the subdialog or applicable set of domain plans being executed have been identified, the system looks for constraints upon these plan steps that have been propagated and inferred from prior interaction. For example, a destination, origin and time constraint may have been previously specified. These constraints are propagated and will also constrain the objects that satisfy those constraints, for example a set of applicable flights. These constraints are then used to refine the set derived from the utterance alone to determine the final set of criteria the rerecognition predictions must fulfil. These predictions are then translated into semantic nets and constraints upon those nets (as illustrated in Figure 1) and used to guide the rerecognition process.

5. Semantic Decoding

We have developed a speech recognition system which uses Recursive Transition Networks as a language model to control word sequences searched for when decoding an utterance [10]. In particular, we use the semantic fragment networks used by the Phoenix parser.

The Phoenix system uses Recursive Transition Networks to represent patterns for semantic fragments. A network specifies those word strings that represent the same concept, and is generated from a semantic grammar. The parser uses the patterns to fill slots in semantic frames. Unlike most RTN systems, our top-level RTN patterns don't generally match entire sentences, but sentence fragments that have a particular meaning. For example, all word sequences that specify a depart location would be a single network, and those specifying an arrive location would be a different network. The utterance "I want to see flights from Boston to denver after 5 pm" would be the concept sequence [list] [select_field] [from_location] [to_location] [depart_time_range]. The system searches for a sequence of concepts, where word sequences constituting concepts are specified by RTNs. Since these RTNs are the same ones used by the parser, it will only produce strings that can be parsed, or nothing at all.

The Sphinx I system [11] was used as the basis for the recognizer. This system uses discrete Hidden Markov Models to represent context dependent phone models. Word models are generated by concatenating the appropriate phone models. A time-synchronous Viterbi beam search is used to match sequences of word models against the input. The original Sphinx system uses bigrams to model the word sequences. We modified the Sphinx recognition search to use RTNs to determine word transitions.

In our two pass system, the original Sphinx system produces the hypothesized word string on the first pass. Our RTN decoder is used for the rerecognition in the second pass.

6. Results

We evaluated the effectiveness of the rerecognition methodology on spontaneous spoken utterances using three official DARPA test sets. Specifically, we compared the performance of the standard Sphinx-I bigram system with the two-pass, RTN-based system. The same lexicon, phone models and word-bigrams were used by both systems.

The systems used a lexicon of approximately 1800 words, including ten non-verbal events. The word-class bigrams were trained on approximately 12000 utterances taken from the DARPA ATIS2 training set, and have a perplexity of approximately 55. There are 79 concept nets. Trigram probabilities of sequences of concepts were also trained on the same set of utterances as the word bigram.

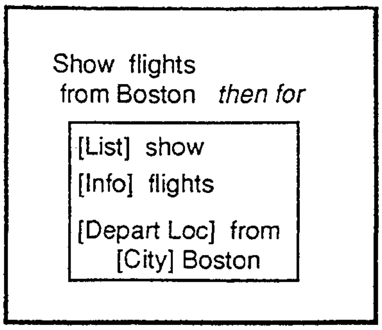

Figure 2: Sample Parser Input and Output

The results show the reductions in utterance error rates that result from using the two-pass RTN system. Utterance error rate measures whether the system correctly interpretted the meaning of the spoken input as reflected in the database queries and answers output by the system's backend. Specifically, the utterance error rate is the percentage of utterances that did not produce the correct answer from the database. In other words, utterance error rates reflect both word error (the sum of word substitution, deletion and insertion percentages) and understanding error (the system's ability to infer the meaning of the recognized words).

Both systems processed all of the dialogs in each of the test sets. The parses from each spoken utterance were passed to our ATIS back end, which parsed the string and produced a response from the database. The DARPA test sets randomly assess performance on a subset of the input utterances, ensuring that "Class X" queries, or those queries for which there is no reference answer, are not included. As shown in Table 1, the numbers of utterances evaluated in the respective subsets were 889, 287 and 76, or a total of 1252 utterances were evaluated.

Tables 1 and 2 break overall error rate down into contextually consistent and contextually inappropriate word recognition errors. The current two-pass system cannot detect contextually consistent word substitutions. Contextual appropriateness is defined in terms of the discourse plans that can be executed at a specific point in time (e.g. clarify, confirm, correct, continue), the objects, attributes and actions available for reference, and the plan steps that are active. For example, if a flight number is misrecognized and substituted for another flight number that filfills the same semantic constraints previously specified in the dialog (i.e. both go to the same place / leave at the same time / serve a meal, etc.) it is considered to be a semantically consistent recognition error and cannot be detected by the semantic, pragmatic and discourse knowledge available in the current two-pass system. As seen in Tables 1 and 2, 41.6% of all recognition errors, are semantically consistent, accounting for a mean 10.4% overall error rate. These contextually consistent errors can only be detected using acoustic confidence metrics (e.g. [8]).

The majority of errors are contextually inappropriate, and it is on these errors that we can measure the sensitivity of the semantic, pragmatic and discourse knowledge and evaluate the strengths and weaknesses of the approach. As seen in Table 1, the system can both detect (*detected errors*) and generate accurate predictions (*correct predictions*) for most of the semantically inconsistent errors. The two pass system generated correct predictions for 88% of the contextually inconsistent errors, correctly predicting semantic content and translating the predictions into a recognition lexicon and grammar. In other words, the system generated accurate content predictions for more than 50% of the total errors. However, the system was only able to correct approximately 50% of the errors it detected, resulting in an overall error rate reduction of roughly 30%.

As seen in Table 2 and described above, the system can correct errors using two methods: knowledge-based constraint alone (% corrected without rerecognition) and rerecognition. In both cases, the system generates a smenatically and pragmatically constrained lexicon and grammar. However, in some cases, the contextual constraints are strong enough to permit an accurate word string to be substituted for a recognition error without performing rerecognition. The system was often able to correct a recognition error using the semantic, pragmatic and discourse level constraints alone. In such cases the context and knowledge-based constraints were enough to select among competing word string alternatives. In fact, the percent of errors the system is able to correct remains roughly equivalent across test sets, even though the relative percentages corrected via rerecognition and by using knowledge-based constraints alone varied. This finding has implications for why rerecognition was not always successful, as discussed below.

Test Set	Error Type	Initial Error	Detected Errors	Correct Predictions	Errors Corrected	Final Error
Nov 92	Bad Context	11.81	10.46	10.01	6.19	5.62
N=889	Total Error	20.58	10.46	10.01	6.19	14.39
Oct 91	Bad Context	22.65	20.91	18.82	9.06	13.59
N=287	Total Error	38.33	20.91	18.82	9.06	29.26
Feb 91	Bad Context	25.00	25.00	21.05	10.53	14.47
N=76	Total Error	38.16	25.00	21.05	10.53	27.63

Table 1: Errors Rate Reductions from Re-recognition

Test Set	% Errors Context Violations	% Errors Consistent	Correct Preds.	Corrected w/o Rerec.	Correct Nets	Rerec.	Total Corrected
Nov 92	11.81	8.77	10.01	2.47	7.54	3.71	6.19
Oct 91	22.65	15.68	18.82	4.18	14.63	4.88	9.06
Feb 91	25.00	13.16	21.05	6.58	14.47	3.95	10.53

Table 2: Breakdown of Correctible Errors

Even though the system was able to correctly generate content predictions for most of the semantically inconsistent errors, it was not able to correct all of the errors. In fact, even though the two-pass system generated accurate, low perplexity, content predictions for the misrecognized substrings, substring re-recognition accuracy was less than 50%. We believe this is a result of two factors. First, it is likely that the misrecognized substrings are more acoustically confusible and harder to recognize than the correctly recognized or re-recognized input. The fact that these word strings were misrecognized initially points to this fact. Also, the relative invarinace in the total percent of semantically inconsistent errors corrected in spite of the fact that some could be corrected on the basis of semantic and pragmatic constraints alone supports this conclusion. Further, there were not significant differences in the perplexities of the prediction sets that resulted in successful vs. unsuccessful re-recognitions. Second, it is possible that there are subtle differences between recognizing a substring and recognizing and entire utterance that we have not considered.

Overall, we were able to significantly enhance recognition using our two-pass system, reducing overall error rates by 30% and generating accurate, low perplexity predictions for virtually all of the semantically inconsistent errors. However, we were not able to correct two types of errors, sematically consistent errors where perfectly acceptible, meaningful misrecognitions were substituted for actual spoken words, and recognition errors caused by highly confusible acoustic patterns.

References

1. Ward, W.H., "Evaluation of the CMU ATIS System", *Proceedings of the DARPA Speech and Natural Language Workshop*, February 1991.
2. Hindle, D., "Deterministic Parsing of Syntactic Non-fluencies", *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, 1983, pp. 123 - 128.
3. Ward, W.H., "The CMU Air Travel Information System: Understanding Spontaneous Speech", *Proceedings of the DARPA Speech and Natural Language Workshop*, June 1990.
4. Ward, W., "Understanding Spontaneous Speech: The PHOENIX System", *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1991, pp. .
5. Young, S.R., Matessa, M., "Using Pragmatic and Semantic Knowledge to Correct Parsing of Spoken Language Utterances", *Eurospeech-91*, 1991.
6. Young, S.R., Hauptmann, A.G., Ward, W.H., Smith, E.T., Werner, P., "High Level Knowledge Sources in Usable Speech Recognition Systems", *Communications of the ACM*, Vol. 32, No. 2, 1989, pp. 183-194.
7. Litman, D. J. and Allen, J. F., "A Plan Recognition Model for Subdialogs in Conversation", *Cognitive Science*, Vol. 11, 1987, pp. 163-200.
8. Young, S. R., "Dialog Structure and Plan Recognition in Spontaneous Spoken Interaction", *Proceedings of the European Conference on Speech Communication and Technology*, ESCA: Paris, London, 1993.
9. Young, S.R., "Use of Dialogue, Pragmatics and Semantics to Enhance Speech Recognition", *Speech Communication*, Vol. 9, No. (5/6), 1990, pp. 551-564.
10. Ward, W.H., Young, S.R., "Flexible Use of Semantic Constraints in Speech Recognition", *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1993, pp. .
11. Lee, K.F., Hon, H.W., Reddy, R., "An Overview of the SPHINX Speech Recognition System", *IEEE Transaction on Acoustics, Speech and Signal Processing*, Vol. ASSP-38, January 1990.