MINDS-II Feedback Architecture: Detection and Correction of Speech Misrecognitions

Sheryl R. Young and Michael Matessa
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

sy@cs.cmu.edu

Functions:	Communication & Perception
Knowledge:	Language
Foundation:	Cognitive
Traditional Area:	Speech Understanding

Abstract

In this paper we describe the architecture and operation of the MINDS-II system. The system analyzes parsed, recognized speech and tries to detect and correct regions containing misrecognitions. The system uses syntactic, semantic, pragmatic and discourse level analyses to both delimit misrecognized input and hypothesize actual spoken content. The heuristics for deriving content hypotheses primarily rely on constraint satisfaction techniques. Hypotheses are expanded into a recursive transition network grammar and are processed with a finite state recognizer. Finally, when multiple phrases are produced in the re-recognition process, each is folded into the entire utterance in order to find which produces the best overall utterance score. Results show this technique reduces speech recognition errors from roughly 38 to 10 percent across multiple test sets.

1. Overview

This paper describes the MINDS-II spoken language system architecture. MINDS-II analyzes parsed, recognized spoken input to identify misrecognized regions of recognized spoken input and send them to be re-recognized using much reduced lexicons and grammars. The dynamically defined lexicons and grammars used for reprocessing are generated by applying constraints derived from syntactic, semantic, pragmatic and discourse analyses of applicable context and prior history. Specifically, we describe the feedback module where parse and recognition errors are detected and content predictions for misrecognized regions of input are derived and then translated into grammars which define the words that can be recognized in the region. We also describe the re-recognition process and how the dynamically defined grammar is used to guide re-recognition. The system is designed such that the feedback component is only called when regular recognition fails.

Like MINDS, MINDS-II is another example of an interactive or integrated architecture for spoken language understanding, where the state of comprehension affects perception. Unlike MINDS, the MINDS-II system is not completely prediction driven. Rather, the feedback component is only called when there are misrecognitions that cannot be corrected locally by the incorporated SOUL system, that uses prior history and semantic and pragmatic consistency. In these cases, MINDS-II generates semantic content hypotheses and uses them for reprocessing the misrecognized regions of input. Results from development test sets are presented.

1.1. THE PROBLEM

Speech recognition and understanding involves deriving conclusions based upon incomplete and frequently inaccurate input. Recognized speech is only probabilistically correct. Misrecognitions are more likely as vocabulary size or the number of accoustically similar items increase. The acoustic properties of spontaneous speech such as stutterring, utterance of partial words, filled pauses and unknown words further complicate the recognition process. Speech recognition errors are made when spoken words are deleted, unspoken words are inserted or when one word is substituted for another. These errors frequently cause the search for further words to become misaligned with actual word boundaries, producing further recognition errors. An examination of misrecognized utterances reveals that misrecognitions are manifested in four ways. The entire utterance can be meaningless; part of the utterance can be semantically inconsistent with the rest of the utterance; the utterance can be semantically consistent internally but be inconsistent with external context; or the utterance can be meaningful both internally and externally. The following misrecognition example is typical of misrecognitions produced by recognizers employing statistical language models: [1, 6] *"WHAT ARE THE FLIGHTS FROM DALLAS TO BOSTON FOR TOMORROW NIGHT OR EVENING"*. This string is recognized as *"WHAT FLIGHTS FROM DALLAS TO BOSTON'S AT ONE NINETY FOUR EVENING"*. Most of the recognition is accurate. Note how errors span regions of input, due to misalignments and the use of word and sound

co-occurrence information. Hence, its important for systems to evaluate recognized strings and determine their accuracy.

Ideally, when inaccuracies are found, systems will identify the errorful regions and correct them. However, it is most difficult to correct a speech misrecognition using a serial model of speech understanding as information flows in only one direction. Serial models apply syntactic, semantic and pragmatic knowledge to the string(s) output by the speech recognizer. Sometimes higher level knowledge can be used to select a completely error-free utterance from a set of most likely recognitions, otherwise recognition errors tend to be propagated through the system. In contrast, interactive models allow later modules to correct the errors made by earlier processes or modules through feedback. Specifically, the history of the interaction and what is known constrains the operation of the speech recognizer.

2. System Overview: Speech Understanding

The MINDS-II system is the analysis and reprocessing portion of the larger CMU spoken language understanding system. The current application is the DARPA air travel information domain (ATIS). The database used in ATIS primarily contains information from the Official Airline Guide. It tells of flight information, aircraft specifications, and ground transportation. Training and test data are gathered for this common task by employing speakers to verbally interact with a database in order to solve one or more randomly assigned problems with predefined goals and preferences. To perform their task, speakers must verbally query an air travel database. The task is designed to elicit natural, unedited, spontaneous speech. Speakers are NOT restricted to complete, unambiguous, answerable, well formed or syntactically correct utterances. They can ask for any information, regardless of whether the request is reasonable or answerable by the information contained in the database. Training and test data are collected by recording both subject utterances and database responses as speakers perform these verbal tasks.

As can be seen in the figure, all input is initially processed by the SPHINX [8, 5, 7] recognition system. The recognizer is Markov model based and uses a statistical bigram language model. Statistical language models allow any word to follow any other word with a given probability. Word transition probabilities are normally computed on categories of words and then normalized by the number of words in a category. [9, 10, 2] Word categories are both syntactic and semantic. For example, nouns are broken down by semantic categories. This version of the SPHINX recognizer is also trained on noise to enable more robust recognition in the face of human and environment noise (e.g. Hmmm, Ahhhh, Ahmmm). [11, 15] The SPHINX recognizer outputs a single scored string of words. Probabilities are also output for each of the components of the string.

The recognized output is then fed to the PHOENIX [12, 13] caseframe parser for speech. PHOENIX is designed to robustly parse spontaneous speech and is specifically designed to deal with spontaneous phenomena including false starts,

mid utterance corrections and phrase repetitions. [4, 11] The PHOENIX system uses a semantic network grammar. Each semantic concept is indexed to one or more nets. The nets themselves are composed of embedded sequences of smaller nets and can be called recursively. The parser forms a beam of all the nets it can match in a recognized string. It then generates sets of interpretations that account for as much of the input as possible. The interpretation accounting for the most input is selected for output. The PHOENIX system interfaces with the database and MINDS-II. MINDS-II takes PHOENIX input and makes corrections when possible. Detected errors that cannot be corrected are marked as not understood.

Figure 2-1: CMU Spoken Language Understanding System

3. MINDS-II

3.1. MINDS-II Overview

The MINDS-II system corrects misrecognitions manifested by some sort of semantic inconsistency. It works by initially identifying regions of a parsed utterances which are misparsed or likely to contain misrecognitions. The system first tries to correct errors without feedback using the SOUL system [16, 17]. The SOUL system is a knowledge intensive reasoning system that performs a fine-grained analysis of recognized, parsed spoken input. It is designed to process spontaneous speech, inclusive of all its dysfluencies, mid-utterance corrections and edits as well as the ambiguous and un-answerable utterances comonly found in the speech of naive system users. The system relies primarily upon pragmatic and semantic knowledge and performs constraint satisfaction and abductive

reasoning. The system interacts with a dialog module which provides inferred speaker intentions (goals and plans), current focus and discourse structure constraints. These enable the system to apply further constraints derived from applicable context. Errors that cannot be uniquely corrected using the SOUL system are considered for re-recognition. These remaining errors are processed by delimiting the suspect portions of the recognized string, generating semantic content predictions and reprocessing the delimited region using a dynamically defined subset of the system grammar and a finite-state speech recognizer.

INPUT: Input to the MINDS-II system is a string recognized using a statistical bigram grammar, candidate parses of the string and the match matrix generated by the PHOENIX caseframe parser.

PROCESSING: The system begin by looking for regions likely to contain errorful parses and misrecognitions. If these errors cannot be corrected locally using constraints from semantics, pragmatics, history and dialog structure, they are considered for reprocessing. Utterances sent for reprocessing must meet the following criteria: The delimited region must be deemed important. This criteria is designed to deal with the frequent occurrence of minor misrecognitions and extraneous spoken input in otherwise correct, coherent utterances. Reprocessing is designed to correct recognition errors which are likely to effect the meaning or interpretation of the spontaneously spoken utterance.

Reprocessing works as follows. First the system identifies and delimits regions of recognized output believed to contain significant errors. Next, it generates a set of meanings for the region, or plausible content predictions. These meanings are generated to be semantically consistent with all applicable context. The meanings are used to dynamically define a grammar for the region. This is accomplished by translating the predictions into corresponding semantic, recursive transition networks previously defined and employed in the PHOENIX system. Re-recognition is accomplished by feeding the identified region and the dynamically defined nets into a finite-state speech recognizer.

OUTPUT: The system outputs parse corrections, corrected recognitions and their interpretations or error codes specifying what it does not understand and why. These corrections are passed back through the parser to the database interface where the parsed, interpretted strings are translated into database queries.

STRENGTHS: The architecture described was designed to capitalize on strengths of both statistical and semantic grammars. Statistical grammars do not have grammatical coverage problems, any known word can follow any other known word. However, unmeaningful and ungrammatical sequences of words can be produced. Semantic grammars guarantee the production of at least meaningful phrases if not meaningful utterances. However, it is difficult to define all possible surface manifestations of the underlying concepts. Hence, they have trouble with insufficient grammatical coverage.

A second strength of this architecture is that is can be used for isolated

utterances as well as dialog lengthed interactions. The difference is only perceptible in the number of hypotheses generated for each region sent for reprocessing. Dialog based knowledge constrains the set of hypotheses.

3.2. Error Detection

Errors are detected that are manifested in the following three ways:

- Identifying regions of input that contain completely incoherent or meaningless information.
- Determining that the various phrases of input are semantically inconsistent with one another.
    - When semantic inconsistencies are found, the system identifies likely misrecognized phrases by trying to put various combinations of phrases together to form a meaningful whole. This whole is constrained by applicable contextual constraints when a dialog lengthed interaction is processed.
- Detecting that the utterance, while semantically consistent, does not conform to the constraints imposed by current focus or reasonable discourse actions given history and active goals and plans.

Occasionally, errors are due to erroneous parsing. In these cases, the system attempts to reinterpret regions of input. When the system reinterprets misparsed input, it then reanalyzes the utterance to determine whether misrecognitions are likely.

Once utterances with dubiously recognized regions are identified, the system performs syntactic, semantic and pragmatic analyses of the region to determine what the region is likely to modify and whether the region could be an meaningless interjection or misrecognized noise. Such regions occur frequently in otherwise meaningful utterances. The first detection heuristic, designed to pick out regions of "trash" picks out many irrelevant recognition errors.

3.3. Hypothesis Generation

Utterances containing identified regions of misrecognized input are analyzed in order to generate content hypotheses, which, when expanded into grammars, are used for re-recognizing the region of input.

Alternative region boundaries are treated separately. In other words, when uncertainty exists as to whether the misrecognized region spans words 1 - 3 or words 1 - 4, separate hypotheses are developed for each region. As will be discussed later, the resolution of region boundaries is determined by computing complete utterance path scores. In this way competing hypotheses as well as competing boundaries can be evaluated using the same metric.

The analysis begins by applying syntactic constraints to determine what role the region plays in the utterance. Because spontaneous speech is input, we allow for phrase repetitions [4] that would not occur in text. The syntactic analysis tells us what known region the misrecognized input is likely to modify.

Semantic and pragmatic constrains are applied next. These usually eliminate alternative syntactic hypotheses. The semantic and pragmatic knowledge is organized as a multi-layered frame based system of hierarchies.[1] Objects and attributes, actions and events, plans and goals make up the four layers of hierarchies employed. Each concept representation has constraints on each of its slots. Two types of constraints are used, single slot value constraints and n-tuple constraints. N-tuple constraints constrain the values of combinations of slots. For example, short range transportation imposes tuple constraints on origin and destination that uniquely distinguish it from long range transportation. Frequently these constraints must be computed from related available information. For example, known distances or associated airports may be used to compute origin or destination constraints. The utterance level evaluation looks for all value constraints that can be placed on the identified region by computing all constraints and the interactions or n-tuple constraints generated by the combined known entities in the utterance. Unlike the initial interpretation analysis, the hypothesis generation phase attempts to find the most general concepts that are consistent with all constraints imposed by the set of identified knowledge base entities in the utterance.

When dialog lengthed interactions are processed, the constraint set derived from utterance analysis is augmented by discourse level constraints. In other words, many possible hypotheses are eliminated.

At the discourse level we track all goals pursued and their current states of activity or abandonment. Information is also available about incompatible goals and combinations of concurrent goals that have failed. Goals are organized such that any compatible combination of goals can be pursued in a dialog [14]. Given this structure and method of representation, any combination of plans and user defined scenarios would not compromise the system. It might not be able to infer all overall goals, but it can identify component parts given spoken input and database responses. The goals are indexed to plans which are themselves organized hierarchically. Plans contain ordering and precondition constraints. They are also important for determining current focus and possible next actions. We also maintain a history stack, indexed by the intentions derived from discourse segments [3]. This tells of related subdialogs, abandoned plans and goals and the current focus. The current focus and state of the world limits

---

[1]The knowledge can best be viewed as a set of planes, each containing multiple hierarchies, where each plane contains information about entities of similar granularity. For example, objects and their attributes are contained within a single plane, where as actions and events are contained within a second plane. This representation lessens the inconsistencies normally found in large reasoning systems.

available referents.

In order to derive discourse level constraints, we first identify the possible purposes of the utterance given all the possible contexts imposed by the utterance level hypotheses. We use active plans, goals and current referents in conjunction with discourse structure rules (Young90, Gros86) to specify possible next actions. The possible next actions can eliminate some of the hypotheses previously generated. This is because the information in the utterance itself enables us to select a "next action" and specify how the utterance as a whole relates to the history of the interaction. Does it continue the current plans and goals, is it a related subdialog, e.g. requesting information about the last answer or request further information about an available referent? Or does is the speaker returning to an abandoned goal or adding an additional goal or constraint? Once we know how the utterance relates to the discourse, we can determine the available referents and impose constraints from prior information. This enables constraints to be propagated by utterance context.

Because this system relies heavily upon coherence and postulates rational speaker beliefs, it is not able to differentiate a misrecognition from from a misstatement or a mistaken belief.

3.4. Hypothesis Expansion

Each of the content hypotheses for a misrecognized region of an utterance is expanded to produce a recursive transition network grammar. Concepts represented in the knowledge base are indexed to the nets which express the concept. The nets themselves may call other nets or be composed of more minor nets. These embedded nets are also activated when a higher level net is called. The system basically looks up each of the concepts and forms a set of nets.

To further constrain recognition, the ways the nets may expand is limited by available referents. For example, if the concept flight numbers is hypothesized, the current context is used to determine available flight numbers or compute them using existing constraints such as origin, destinations, time, etc. Hence, instead of allowing any possible one to four number string, the number strings are limited to specific sequences.

3.5. Signal Reprocessing

The nets and their expansion limitations are used to re-recognize given regions of speech using a finite state recognizer. This means that as each 10 msec region of speech is processed, we look at each state to determine whether its still in a word or has reached the end of a word. When an end of word state is reached, we perform a grammar lookup to determine all the words which could possibly follow it. The recognizer prunes all paths whose scores fall below a certain level and outputs path scores for all reasonable scoring strings of words.

To select a best interpretation, we do not just use the best scoring path. Rather,

we retain all the path scoring information from the original recognition and fold the various surviving path scores into the entire string. The path that produces the best overall score is selected.

When the boundaries of a segment are in question, we produce two sets of hypotheses and process the overlapping regions separately. To determine which boundary was accurate, we fold the output string produced by both re-recognitions into the overall recognized string. Of course, the portions of the initial string substituted for the re-recognized alternatives is different for each possible boundary. By folding in all output strings, we can directly compare overall utterance scores with one another.

4. RESULTS

To assess potential effectiveness of MINDS-II architecture we assessed the system's ability to detect inaccurate recognitions and to generate correct hypotheses for the inaccurate regions. We also measure grammatical coverage for the nets selected and obtained overall system performance. We tested our system on three separate developmental test sets provided by the DARPA community. The test sets require that all utterances be processed and recognized. Many separate dialogs are included. You must use your own database response to update your representations of current state of the world. The actual utterances the test sets evaluate and score are only a portion of the utterances provided for processing. Roughly, the test sets are evaluated on 50% of the utterances. The accuracy evaluation is performed automatically using software provided by NIST. The input to the evaluation program is the database output produced in response to processing an utterance. In this way, context dependent interpretations can be evaluated. The three test sets below evaluated 75, 198 and 55 utterances, respectively.

Presented below are two separate analyses of the same data. The first analysis is included to provide insight in the effectiveness of the heuristics employed by the system. The second analysis shows the overall effectiveness of the MINDS-II system by contrasting performance with PHOENIX system alone.

Speech Recognition Accuracy						
Test Set	Initial % Error	Errors Detected	Corrected without Feedback	Correct Hypotheses	Corrected	Error Rate
1 (76)	38	33	13	8	8	17
2 (198)	30	17	0	13	12	18
3 (55)	38	20	2	16	16	22

Detection ability places an upper limit on potential accuracy enhancement. What is detected includes incorrect parses, incorrect recognition, unknown words and unknown concepts. Detected items that cannot be corrected become "No Answer" items for official DARPA scoring. This means that the system knows it has not been able to understand or accurately parse the input. Roughly 60% to 80% of all errors are detected.

Some misrecognitions can be corrected without feedback when history and context dictate a unique candidate. Usually this occurs with alphanumeric substitutions such as flight numbers or any other concepts with alphanumeric values. Similarly, some inaccurate parses can be corrected. Errors corrected without feedback further limit the maximum number of correctable misrecognitions. As seen above, the number of errors correctable without feedback varies considerably across test sets.

There are three factors which determine whether errors can be corrected via reprocessing. First, the error must be due to misrecognition, not erroneous parsing. Operationally, we define this as one or more (potentially overlapping) regions of input believed misrecognized. Second, we must correctly generate a meaning hypothesis for the misrecognized region. Generating accurate meaning hypotheses are a precondition for accurate re-recognition of a previously misrecognized region of spoken input, because the meaning hypotheses are used to define the new, reduced search space for words. To assess whether correct hypotheses are generated, or *correct hypotheses*, we compare standardized utterance transcriptions with the words and phrases associated with the concepts hypothesized for each denoted region of input. This result directly measures the reliability of our heuristics.

Third, the grammar nets which expresses the hypothesized meaning must have a representation for the actual words spoken, or adequate grammatical coverage. There are two ways in which a grammar may lack coverage. First, the system may lack lexical coverage for some of the words used to express the hypothesized concept or group of concepts. Second, the system may lack grammatical representation for this particular grouping or ordering of words which express the hypothesized content. However, in these test sets, very few re-recognitions suffered from inadequate grammatical coverage. Since the hypotheses expand into a highly restricted grammar, actual recognition is almost assured when sufficient grammatical coverage exists.

Speech Results				
Test Set	System	% Correct	% No Answer	% Wrong
1 (76)	PHOENIX	62	0	38
	MINDS-II	83	12	5
2 (198)	PHOENIX	70	0	30
	MINDS-II	82	5	13
3 (55)	PHOENIX	62	0	38
	MINDS-II	80	2	18

References

[1] Bahl, L. R., Jelinek, F., Mercer, R.
A Maximum Likelihood Approach to Continuous Speech Recognition.
IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-5(2):179-190, March, 1983.

[2] Derr, A., Schwartz, R.
A Simple Statistical Class Grammar for Measuring Speech Recognition Performance.
In *DARPA Speech Recognition Workshop*. October, 1989.

[3] Grosz, B. J. and Sidner, C. L.
Attention, Intentions and the Structure of Discourse.
Computation Linguistics 12:175-204, 1986.

[4] Hindle, D.
Deterministic Parsing of Syntactic Non-fluencies.
In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pages 123 - 128. 1983.

[5] Huang, X.D., Lee, K.F., Hon, H.W., Hwang, M.Y.
Improved Acoustic Modelling with the SPHINX Speech Recognition System.
In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 345-348. 1991.

[6] Jelinek, F.
Self-Organized Language Modeling for Speech Recognition.
Readings in Speech Recognition.
In A. Waibel & K.F. Lee,
Morgan Kaufmann, 1990, pages 450-506.

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

[8] Lee, K.F., Hon, H.W., Reddy, R.
An Overview of the SPHINX Speech Recognition System.
IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-38, January, 1990.

[9] Stern, R.M., Ward, W.H., Hauptmann, A.G., Leon, J.
Sentence Parsing with Weak Grammatical Constaints.
In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 380-383. IEEE, 1987.

[10] Ward, W.H., Hauptmann, A.G., Stern, R.M. and Chanak, T.
Parsing Spoken Phrases Despite Missing Words.
In *IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1988.

[11] ward, W.H.
Modelling Non-Verbal Sounds for Speech Recognition.
In *DARPA Speech Recognition Workshop*. October, 1989.

[12] Ward, W.H.
The CMU Air Travel Information System: Understanding Spontaneous Speech.
In *DARPA Speech Recognition Workshop*. June, 1990.

[13] Ward, W.H.
Evaluation of the CMU ATIS System.
In *DARPA Speech Recognition Workshop*. February, 1991.

[14] R. Wilensky.
Planning and Understanding.
Addison Wesley, Reading, MA, 1983.

[15] Wilpon, J., Rabiner, L., Lee, C.-H., Goldman, E.
Automatic Recognition of Keywords in Unconstrained Speech using Hidden Markov Models.
IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-38(11):1870-1878, 1990.

[16] Young, S.R.
Using Semantics to Correct Parser Output for ATIS Utterances.
In *DARPA Speech Recognition Workshop*, pages 106-111. February, 1991.

[17] Young, S.R., Matessa, M.
Using Pragmatic and Semantic Knowledge to Correct Parsing of Spoken Language Utterances.
In *Eurospeech-91*. 1991.

Table of Contents

List of Figures