ISCA Archive
http://www.isca-speech.org/archive

ITRW on Error Handling in Spoken
Dialogue Systems
Château d'Oex, Vaud, Switzerland
August 28-31, 2003

Using Grammatical Analysis to Detect Misrecognitions

Teresa Zollo

Department of Computer Science
State University of New York College at Geneseo
zollo@cs.geneseo.edu

Abstract

In systems that use grammatical analysis rather than concept spotting to accomplish natural language understanding, the presence or absence of the top-level constituent "turn" can be used to reliably detect whether the user's speech was misrecognized. In this paper, a description of the structure of well-formed spoken turns in practical human-computer dialogue is given. We explain how that description of turns can be encoded the context-free grammar rules used by a parser, and how the result of the parser's analysis can be used as a basis for detecting misrecognitions. We provide the results of an evaluation of this error detection strategy in the TRIPS-Pacifica domain showing 92.1% accuracy in classifying speech recognition hypotheses as correct or erroneous, an improvement of 18.2 percentage points above the majority-class baseline.

1. Introduction

A key technology enabling the deployment of spoken dialogue systems is automatic speech recognition (ASR). The basic functionality of an ASR system is to take as input the acoustic signal produced when a speaker articulates a sequence of words, and generate a string of words that corresponds to the words spoken. Typically in a spoken dialogue system, the output of the automatic speech recognition subsystem is the input to the natural language understanding component in a pipelined fashion.

Unfortunately, user-independent conversational speech recognition is still far from perfect. Results from NIST's 2001 evaluation of large vocabulary conversational speech recognition systems show that the best such systems recognize approximately 80% of the words correctly [1].

As a result of this poor level of performance, a major challenge facing developers of spoken dialogue systems is to design systems that can perform robustly in the face of misrecognitions. Current spoken dialogue systems operate within narrow domains, and many perform natural language understanding by "concept spotting": picking out words and phrases in the ASR hypotheses that are salient to particular tasks that the system can perform. Such systems do not require a full syntactic analysis of what the user said; they can accomplish their mission by recognizing only a few key phrases. Thus, in these systems, speech recognition hypotheses do not need to be perfect as long as those salient words and phrases are correctly recognized.

However, picking out phrases without regard for the larger syntactic context in which they occur can easily lead to error. Even if the phrase is a correctly recognized constituent of the user input, its significance in the larger context may be entirely different from its conjectured significance in isolation. Thus, a phrasal approach to interpretation that simply ignores problematic segments is apt to go astray, by failing to exploit the constraints supplied by the larger syntactic context, and the contributions to meaning from that context.

Table 1 shows an interaction taken from the TRAINS-95 evaluation corpus [2], where the system, by relying on concept spotting to provide robustness, misunderstands the user's intention. In the example dialogue segment shown in Table 1, ASR hypotheses are preceded by HYP, the words actually spoken by the user are preceded by REF, and the system responses are preceded by S. In this excerpt, the user is unable to rectify the misunderstanding, becomes frustrated, and finally ends the dialogue without completing the task.

U:	HYP:	**I'd d nine s i** WANT TO GO FROM CLEVELAND TO BOSTON
	REF:	I **do not** WANT TO GO FROM CLEVELAND TO BOSTON
S:		I need help choosing a route from to Cleveland Boston.
U:	HYP:	DO NOT **descending** THE TRAIN FROM CLEVELAND TO BOSTON
	REF:	DO NOT **send** THE TRAIN FROM CLEVELAND TO BOSTON
S:		OK I need help choosing a route from Cleveland to Boston.
U:	HYP:	**I'd too now i** WANT TO GO FROM CLEVELAND TO BOSTON
	REF:	I **do not** WANT TO GO FROM CLEVELAND TO BOSTON
S:		I can't find a route to get from Cleveland to Boston.
U:	HYP:	I'M DONE

Table 1: Example of Misunderstanding from TRAINS-95

The dialogue segment in Table 1 clearly demonstrates the danger in simply disregarding substrings in an ASR hypothesis that do not fit into the system's expectations. In the first exchange, the system disregarded the words that correspond to "I do not" yet accepted the string "I want to go from Cleveland to Boston", In doing so, the system formed an understanding that was the *exact opposite* of what the user intended.

Misunderstandings caused by poor speech recognition pose a serious problem for dialogue systems since they are difficult to detect and correct, and can result in poor overall performance. This paper describes an experiment in detecting misrecognitions as ASR hypotheses are being processed by the natural language understanding component of a spoken dialogue system. By detecting problems with the ASR hypothesis, misunderstandings can be averted. Detecting ASR errors enables the dialogue system to adapt its dialogue behavior appropriately.

When input from the ASR system is identified as faulty, the dialogue system may decide to enter into a repair subdialogue, selectively verify its interpretation, transition to a strictly system-initiated dialogue style [3], switch from the automated system to a human attendant [4], or try to guess at what was actually said based on statistically-driven expectations and phonetic similarity with the ASR hypothesis [5].

As the tasks performed by spoken dialogue systems become more complex and general, the system will require a more thorough analysis of user input, rendering many current robustness strategies ineffective. Our experimental system, TRIPS-Pacifica, performs a grammatical analysis using a general-purpose grammar to form an interpretation of the ASR hypothesis.

In spoken dialogue, the participants alternate in the roles of speaker and listener, and in making contributions to the dialogue during their turn as speaker. Many times, speakers make several distinct contributions to the dialogue within a single turn. The error detection strategy we investigated relies on the grammar to describe plausible user turns, where a "turn" is the sequence of words uttered by a dialogue participant during a consecutive period in which he or she has is making contributions to the dialogue without seeking contribution from an interlocutor. Since ASR hypotheses are supposed to represent user turns, ASR hypotheses that do not conform to the grammar's description of user turns are classified as erroneous.

To evaluate how well our strategy for ASR error detection works in practice, we implemented them in the context of a particular dialogue system, TRIPS-Pacifica, developed at the University of Rochester. TRIPS-Pacifica functions as an assistant to a human who has been given the task of planning the evacuation of the fictitious island of Pacifica [6].

The speech recognition component used by TRIPS-Pacifica is Sphinx-II, a continuous, speaker-independent recognizer developed at at Carnegie Mellon University [7]. The acoustic models were trained on a combination of data from the Air Travel Information System (ATIS) and TRIPS-Pacifica, and the lexicon of approximately 1800 words was tailored for the TRIPS-Pacifica domain. Sphinx-II was configured to provide a single best hypothesis; confidence scores were not provided. The TRIPS-Pacifica parser is a bottom-up chart parser that uses a constraint-based grammar [8].

The context-free grammar used by TRIPS-Pacifica is somewhat different from the grammars used by other natural language parsers in that the top-level constituent is a *turn* rather than a *sentence*. A turn differs from a sentence in that it may consist solely of a fragment (phrasal constituent), or it may consist of a combination of sentences and fragments.

Spoken turns frequently do not form the complete sentential units we typically find in text, and often a speaker performs several distinct speech acts within a single turn, each of which may either be a full sentential unit or a fragment. Most existing dialogue systems simplify the problem of understanding by assuming that a user's turn will contain a single contribution in the form of a sentence or phrasal unit. As dialogue systems evolve and interactions become more natural (that is, become closer to interactions between human conversants), turns are likely to contain multiple contributions more and more frequently. In the corpus of dialogues collected in our experimental domain for the purpose of evaluating our error detection strategy, 14% of turns consisted of multiple contributions.

People engage in conversations to achieve goals, and the act of saying something is intended by a speaker to contribute toward those goals. In casual conversation, the goals at any given time may not be well-defined, resulting in a lack of structure, predictability and fluency. The model of spoken turns presented in this paper applies to dialogues in which humans are interacting with computer systems to accomplish a well-defined goal. Practical dialogues are a simplification of general conversation since the type of speech used during practical conversations tends to be more formal, the vocabulary used is more constrained, and speakers tend to formulate their contributions more carefully. Human-computer interaction provides further simplification, since the issue of turn-taking, at least on one side, is under the control of the computer system.

2. Context-free Grammars for Natural Language

For the most part, natural language syntax can be described by a *context-free grammar* (CFG). In general, a CFG is a 4-tuple composed of:

1. an *alphabet* consisting of all the terminal symbols of the language
2. a set of *nonterminal symbols* of the language
3. a special nonterminal symbol called the *start symbol*
4. a set of rules, called *grammar rules* (also commonly referred to as production rules and rewrite rules) that describe how the terminal and nonterminal symbols can be combined to form other nonterminals.

In the case of spoken natural language, the alphabet is the set of words in the language (L). In a parsing component of a spoken dialogue system, this set of words is some subset of L, specified in the lexicon. The lexicon in a spoken dialogue system will include commonly-used words and terms that are likely to be used in the domain of discourse.

The set of nonterminal symbols includes lexical categories such as noun and verb and higher-level categories such as noun phrase and verb phrase.

In grammars for spoken natural language, the nonterminal symbol used as the start symbol, or the highest-level constituent formed, is often sentence. This is problematic since the sequences of words spoken often do not form sentential units. Having sentence as the top-level constituent is a holdover from earlier natural language parsers that processed text rather than speech.

We need to characterize spoken language in terms of a different top-level constituent or start symbol than that used for text. In this paper, we borrow from philosophy of language and use the term "speech act" to refer to the words that form a distinct contribution, such as an acknowledgment or command. The term "turn" will refer to all the words spoken during a single turn as speaker, which may be composed of multiple speech acts. Based on these definitions, a syntactic analysis of the string "OKAY GOOD NOW FLY THE HELICOPTER FROM EXODUS TO ABYSS AND PICK UP THOSE PEOPLE" would show a single turn constituent comprised of three speech acts:

1. the acknowledgment OKAY
2. the evaluation GOOD
3. the request FLY THE HELICOPTER FROM EXODUS TO ABYSS AND PICK UP THOSE PEOPLE

Thus, speech acts are made up of words and turns are made up of speech acts. Note that speech acts can be realized as a single word, such as "hello" or "okay", and many times turns consist of a single speech act.

2.1. Context-Free Grammar Rules for Spoken Turns

The purpose of any grammar is to describe a language in such a way that all strings belonging to the language fit the description, and strings not belonging to the language do not fit the description. Grammars for natural language can only approximate this goal; there is always a tradeoff between being too permissive and allowing strings that are not part of the language on one hand, and being too restrictive and disallowing some strings belonging to the language on the other.

In his famous example "COLORLESS GREEN IDEAS SLEEP FURIOUSLY", Noam Chomsky demonstrates that syntax is independent of semantics. Chomsky's sentence is *syntactically* correct but *semantically* incoherent. Although Chomsky was at pains to show that meaningfulness is not a matter of syntax, many natural language parsers capture semantic constraints in their context-free grammar rules. Typically, natural language parsers used in dialogue systems attribute semantic features to linguistic constituents and enforce semantic consistency by grammar rules that require unification of semantic features. For example, in order to prohibit semantically anomalous sentences such as Chomsky's example, adjectives such as "green" have a feature that specifies they can only be applied to physical objects, and nouns such as "ideas" have a feature specifying that they are concepts (as opposed to physical objects). The grammar rule that forms a noun phrase from an adjective and a noun would then ensure that the features for the adjective and the noun are consistent (that is, that they "unify"). Of course, it is extremely difficult to capture all the subtleties of natural language semantics in the feature system of a grammar; all current natural language grammars are imperfect.

In developing a grammar for spoken natural language, we make the following claim: We can capture much of the regularity of turn structure, in particular the way in which speech acts are arranged within turns, via context-free rules, despite the fact that the order of speech acts in a turn would normally be regarded as a matter of pragmatics. To demonstrate that this is the case, consider the turn "OKAY GOOD NOW FLY THE HELICOPTER FROM EXODUS TO ABYSS AND PICK UP THOSE PEOPLE".

This turn is an actual user turn from the TRIPS corpus, and is an example of a well-formed, multiple-act turn. In this turn, the speaker acknowledges the other dialogue participant's previous contribution, evaluates its content, and then issues a request. But consider the turn that results from rearranging the order of the speaker's contributions: "NOW FLY THE HELICOPTER FROM EXODUS TO ABYSS AND PICK UP THOSE PEOPLE GOOD OKAY"

Our claim is that although each speech act within the latter turn is syntactically and semantically well-formed, the turn as a whole is infelicitous for pragmatic reasons. One aspect of our grammar for spoken turns in practical dialogue is that it restricts the order in which speakers perform speech acts within a turn.

2.2. Speech Act Classification

For a turn to be coherent, speech acts that address outstanding dialogue obligations, such as acknowledgments and repair requests, should appear early in a turn, and speech acts that create new dialogue obligations, such as questions, should appear later. The grammar classifies speech acts as one of the following:

initial acts speech acts that must appear in the initial part of the turn

mid acts acts that must appear after initial acts and before final acts (if any are present in the turn)

final acts acts that must appear at the end of a turn

In the original TRIPS-Pacifica grammar, utterances were the highest-level constituent formed. The grammar attributed a feature called "speech act" to utterances, to describe the parser's preliminary determination of the illocutionary force of the utterance. The value of the speech act feature for a given utterance could have been based on the syntactic structure of the utterance, on the presence of cue words such as "please", or could have been specified in the lexicon for certain one-word utterances such as "yes". Table 2 shows all possible speech act values assigned by the TRIPS parser, and their correspondence to our classification scheme for speech act types.

TRIPS **Speech Act**	**Position**	**Example**
ACCEPT	INITIAL	OKAY
APOLOGIZE	INITIAL	SORRY
CLOSE	MID	BYE
CONFIRM	INITIAL	YES
EVALUATION	INITIAL	GOOD
EXPRESSIVE	INITIAL	THANKS
HOLD	INITIAL	ACTUALLY
GREET	INITIAL	HI
HOW-QUESTION	FINAL	HOW CAN I MOVE THE PEOPLE FROM CALYPSO TO BARNACLE
NOLO-COMPRENDEZ	INITIAL	PARDON
REJECT	INITIAL	CANCEL THAT
REQUEST	MID	SEND A HELICOPTER FROM DELTA TO EXODUS
SUGGEST	MID	WHY DON'T YOU TAKE THE HELICOPTER FROM DELTA TO EXODUS
TELL	MID	THE ROAD IS OUT BETWEEN CALYPSO AND OCEAN BEACH
WH-QUESTION	FINAL	WHERE ARE THE VEHICLES
WHAT-IF QUESTION	FINAL	WHAT IF WE TAKE THE HELICOPTER FROM DELTA TO EXODUS
WHY-QUESTION	FINAL	WHY IS THE SHORTEST ROUTE THROUGH CALYPSO OVERLOOK
YN-QUESTION	FINAL	AM I DONE

Table 2: Speech Acts Assigned by the TRIPS Pacifica Grammar

The many-to-one correspondence between TRIPS-Pacifica speech acts and the three speech act categories in our scheme makes the interface between our rules to form turn constituents and the underlying TRIPS-Pacifica grammar straightforward. We simply form initial act, mid act, and final act constituents from utterance constituents having certain speech act feature values as indicated in Table 2.

Our grammar rules allow for sequences of initial act and mid act utterances within a single turn, but only one final act is allowed. The rationale behind this restriction is that once a speaker creates a strong obligation for their partner to respond, by asking a question, the speaker will immediately end the turn so that the obligation can be fulfilled.

2.3. Fragments

Another issue in spoken language that does not tend to come up in natural language text is that of *fragments*. Speakers will often use fragments as a kind of "shorthand" way of making a contribution when they are confident that their intention will be understood.

Any grammar for spoken language needs to accommodate fragments. However, there seem to be significant restrictions on the use of fragments in spoken turns (at least in practical human-computer dialogue), which we reflected in our grammar rules to form turn constituents. The three restrictions on fragments that our rules enforce are:

1. A fragment must constitute a full phrasal unit.
2. A turn will contain at most one fragment.
3. If a fragment is present in a turn, it must be either at the very beginning or the very end of the turn.

Although we do not attempt to assign a speech act designation to fragments in our grammar, we would expect that fragments being used to fulfill outstanding dialogue obligations will appear in the initial position, and fragments that create new dialogue obligations will appear in the final position.

2.4. Disfluencies

Although the number of disfluencies in human-computer interactions is low compared to the number in human-human conversation, they do occur occasionally. For example, there were a few examples of speakers repeating the same word twice (consecutively) in the evaluation corpus. The issue of repeated words can be easily solved by including grammar rules that allow consecutive occurrences of a word or words to be condensed into a single occurrence.

Much of what at first glance appears to be fragmented speech is in fact the speaker making a mid-turn repair. There has been much research on the problem of accounting for speech repairs in computational systems. Some researchers have proposed a preprocessing phase prior to parsing to identify speech repairs [9], while others have used special rules in the parser's grammar to account for speech repairs [10]. The error detection processing described in this paper assumes that fragments resulting from self-repair have been previously dealt with by one of these proposed methods, and does not explicitly deal with them. The few speech repairs observed in the evaluation corpus are consistent with the findings of the previous work mentioned above. All of the repairs have a predictable structure that make them suitable for automatic processing.

One other type of disfluency that was fairly common in the evaluation corpus was the use of hesitation words, such as "UM". Hesitation words in the ASR hypothesis are simply ignored by the phrasal and sentential grammar rules. Thus, they can appear anywhere within a turn without affecting the error detection processing.

2.5. Implementation of the Turn Rules

We added fourteen rules to the TRIPS-Pacifica grammar to create turn constituents from combinations of fragments, sequences of initial and mid speech acts, and final speech acts. Note that every rule to form a turn constituent also has the start-of-turn marker as the first right hand side constituent and the end-of-turn marker as the last right hand side constituent. This guarantees that a turn spans the entire ASR hypothesis.

Table 3 shows the fourteen rules to form turn constituents, and indicates how often each rule was used in the evaluation corpus. (Note: The start-of-turn and end-of-turn markers are omitted from the table.)

Turn Rule RHS Constituents	# in TRIPS Corpus
empty	0
FRAGMENT only	26
FRAGMENT,INITIAL	1
FRAGMENT, MID	1
FRAGMENT, MID, FINAL	0
FRAGMENT, FINAL	0
INITIAL, FRAGMENT	1
INITIAL only	30
INITIAL, MID	22
INITIAL, FINAL	18
INITIAL, MID, FINAL	2
MID only	115
MID, FINAL	2
FINAL only	202

Table 3: Distribution of Turn Types in the TRIPS Corpus

Not only does our turn grammar describe 98% of the actual user turns from the evaluation corpus, it also does an excellent job of identifying strings that *do not* represent user turns. Both of these aspects are critical to the performance of our misrecognition detection strategy, which relies on the parser and grammar to recognize ASR hypotheses as either well-formed turns or not. Most of the restrictive power in our grammar comes from our restrictions on exactly what constitutes a fragment and the number of fragments in a turn. In most of the erroneous ASR hypotheses, we find words or substrings that are not part of a speech act and do not form full phrasal units. In other erroneous ASR hypotheses, we find multiple fragments. The restrictions that we place on the position of fragments in a turn and the position of speech act types within a turn do not contribute greatly to the grammar's restrictive power, suggesting that perhaps these distinctions are unnecessary.

3. Evaluation

The TRIPS-Pacifica spoken dialogue system accomplishes natural language understanding by performing a bottom-up chart parse of the ASR hypothesis, and then analyzing the constructed chart to determine the "best" analysis, which is then passed to the dialogue manager.

Our strategy for detecting misrecognized speech recognition hypotheses is a very simple one that uses the parser as a *recognizer*. A recognizer is a program that outputs a determination of whether or not the input string conforms to a specific grammar. Our turn model is encoded in context-free grammar rules (to form turn constituents) which are then added to the set of grammar rules already used by the TRIPS-Pacifica system. After constructing the chart for a particular input string, our classifier simply checks whether a turn constituent exists in the chart. If a turn constituent is present, the classifier guesses that the ASR hypothesis is correct; otherwise, the classifier guesses that the ASR hypothesis is misrecognized. Figure 1 shows pseudocode for a simple boolean function that returns an indication of whether or not the last ASR hypothesis parsed should be classified as a misrecognition. The function to detect whether or not the ASR hypothesis was misrecognized is called after the chart has been constructed by the parser.

```
function is–probable–misrecognition returns boolean
    if (no TURN constituent was formed for this string) then
        return TRUE
    else
        return FALSE
    end if
end is–probable–misrecognition
```

Figure 1: Pseudocode for ASR Misrecognition Detection

To evaluate the performance of our error detection processing, a set of 429 ASR hypotheses for turns in the TRIPS-Pacifica domain was collected as users were actually using the system to accomplish a task. To create a reference set of "correct" results, the Sphinx-II hypotheses were compared to transcriptions of the turns that were created manually by an experimenter listening to the audio files that were stored as part of the data collection. If a hypothesis was not identical to the reference transcription, then it was tagged as being misrecognized. According to the reference answers, 317 of the 429 ASR hypotheses (73.9%) were misrecognized.

During the evaluation of the error detection processing, we fed each ASR hypothesis into the TRIPS-Pacifica parser using the "online" feature of the parser described in [8]. No modifications were made to the parser itself, but the grammar rules used by the parser were augmented to include our rules to construct turn constituents. After the parser constructed the chart, the evaluation procedure invoked the error detection code and recorded the output. The output of the automatic processing was then compared to the set of reference answers to determine the results reported in Section 3.1.

3.1. Results

We evaluate the ASR error detection processing by reporting accuracy for the task of classifying ASR hypotheses as correct or misrecognized. We also report precision and recall metrics for the task of detecting misrecognitions. We provide 95% confidence intervals for each of the results.

		Classifier Results	
		Event	$\overline{\textbf{Event}}$
Ref. Tags	**Event**	Correct Pos	Incorrect Neg
	$\overline{\textbf{Event}}$	Incorrect Pos	Correct Neg

Table 4: Confusion Matrix for Computing Classifier Performance

For each hypothesized turn from the speech recognizer, the algorithm classifies the hypothesis as correct or misrecognized. Table 5 is a confusion matrix showing the number of correct and incorrect classifications, which are used to compute performance metrics for our classifier.

		Classifier Results	
		ASR Error	$\overline{\textbf{ASRError}}$
Ref. Tags	**ASR Error**	295	22
	$\overline{\textbf{ASRError}}$	12	100

Table 5: Confusion Matrix for Grammar-Based Method

A baseline classifier for the task of tagging ASR hypotheses as correct or misrecognized will always tag hypotheses as misrecognized, given the majority class from our test set. The confusion matrix showing the results of a majority class classifier is shown in Table 6.

		Classifier Results	
		ASR Error	$\overline{\textbf{ASRError}}$
Ref. Tags	**ASR Error**	317	0
	$\overline{\textbf{ASRError}}$	112	0

Table 6: Confusion Matrix for Baseline Method

Accuracy is the ratio of number of times the algorithm produces the correct classification (correct positives + correct negatives) to the number of times the event actually occurred (correct positives + incorrect negatives + incorrect positives + incorrect negatives).

The first line of Table 7 shows the accuracy achieved by our strategy of having the parser form turn constituents as the highest level constituent, and basing our classification on whether there is a turn constituent in the resulting chart. The next line shows the accuracy of always guessing the majority class (that the ASR hypothesis contains an error); this is the baseline against which our strategy is compared.

Strategy Used	**Accuracy (%)**
Test If Turn Constituent Formed	92.1 ±2.55
Baseline (Majority Class)	73.9

Table 7: Accuracy for the Classification Task

Precision is the ratio of the number of times an event is correctly identified (correct positives) to the number of times the event is identified: whether correctly or incorrectly (correct positives + incorrect positives).

Recall is defined to be ratio of correct identifications of an event (correct positives) to the number of times the event actually occurred (correct positives + incorrect negatives).

3.2. Discussion

Our error detection processing implicitly assumes that any communication problem is caused by a speech recognition error.

Strategy Used	Precision (%)	Recall (%)
Test If Turn Constituent Formed	96.1 ±1.83	93.1 ±2.36
Baseline (Majority Class)	73.9	100

Table 8: Precision and Recall for Detecting Misrecognitions

Therefore, the performance of our classifier is closely tied to the coverage of the underlying sentential and phrasal grammar rules and lexicon, as well as processing to handle disfluencies such as speech repairs and repeated words. The error detection code is extremely reliable in predicting that errors are errors; the variance is seen when processing correct hypotheses that were misunderstood by the natural language understanding component for some other reason.

Of the 22 misrecognized hypotheses that were predicted to be correct, 10 are pragmatically equivalent to the actual user turn. By "pragmatically equivalent" we mean that the errors were so minor that the precise meaning and intention were unchanged.

Our classifier tagged twelve erroneous ASR hypotheses as being correctly recognized. Some of these turns look like they could be correct, and others were mistagged because the grammar is sometimes too permissive. The analyses of these turns created by the parser were not consistent with the intentions of the speaker. For turns such as these, we need to create strategies in the dialogue manager to recognize that a misunderstanding has occurred. For more discussion, and examples of these errors, the reader is referred to [5].

4. Conclusions and Future Work

In dialogue systems that perform a grammatical analysis of the ASR hypothesis, treating the top-level constituent as a turn and then checking to make sure that the ASR hypothesis is a well-formed turn, is a reliable method for detecting misrecognitions, though we would like to verify our findings on more test data from a variety of domains. Much of the work in detecting misrecognitions has been motivated by applications that automate the functions of human operators, such as providing telephone operator services or answering public transportation timetable queries. In these systems, dialogues found to be problematic can simply be transferred to a human attendant. The motivation for detecting errors in the TRIPS-Pacifica system is somewhat different. We are currently experimenting with actually correcting errors in turns deemed to be misrecognized. Furthermore, when the natural language understanding component detects a misrecognition, that information can be passed to the dialogue manager, which can then make judicious use of repair subdialogues and verification requests, to avoid the types of misunderstandings shown in Table 1.

5. Acknowledgments

The work reported here was conducted at the University of Rochester Computer Science Department and was supported by ONR research grant N00014-01-1-1015, DARPA research grant F30602-98-2-0133, and NSF grants EIA-0080124 and IIS-0082928.

6. References

[1] Martin, A. and Przybocki, M., "The 2001 NIST Evaluation for Recognition of Conversational Speech Over the Telephone", Proceedings of the 2001 Large Vocabulary Conversational Speech Recognition Workshop, 2001.

[2] Sikorski, T. and Allen, J., "TRAINS-95 System Evaluation", TRAINS Technical Note 96-3, Computer Science Department, University of Rochester, 1996.

[3] Litman, D., Hirschberg, J., and Swerts, M., "Predicting Automatic Speech Recognition Performance Using Prosodic Cues", Proceedings of the North American Meeting of the Association for Computational Linguistics, 2000.

[4] Walker, M., Langkilde-Geary, I., Wright-Hastie, H., Wright, J., and Gorin, A., "Automatically Training a Problematic Dialogue Predictor for a Spoken Dialogue System", Journal of Artificial Intelligence Research, 12, 2001.

[5] Zollo, T. "Detecting and Correcting Speech Recognition Errors During Natural Language Understanding", Ph.D. Thesis, Computer Science Department, University of Rochester, 2003.

[6] Ferguson, G. and Allen, J., "TRIPS: An Intelligent Integrated Problem-Solving Assistant", Proceedings of AAAI-98, 1998.

[7] Huang, X., Alleva, F., Hon, H., Hwang, M., Lee, K., and Rosenfeld, R., "The Sphinx-II Speech Recognition System: An Overview", Computer, Speech and Language, 1992.

[8] Allen, J. "The TRAINS-95 Parsing System: A User's Manual", TRAINS Technical Note 95-1, Computer Science Department, University of Rochester, 1995.

[9] Heeman, P. "Speech Repairs, Intonational Boundaries and Discourse Markers: Modeling Speakers' Utterances in Spoken Dialog", Ph.D. Thesis, Computer Science Department, University of Rochester, 1997.

[10] Core, M. "Dialog Parsing: From Speech Repairs to Speech Acts", Ph.D. Thesis, Computer Science Department, University of Rochester, 1999.