A Model of Speech Repairs and Other Disruptions

Mark G. Core and Lenhart K. Schubert
Department of Computer Science
University of Rochester
Rochester, NY 14627
mcore,schubert@cs.rochester.edu

Abstract

Most dialog systems ignore the problem of speech repairs and editing terms (*um*, *uh*, etc.) or use preprocessing techniques to eliminate them from the input. These systems also typically enforce a strict turn-taking protocol that does not allow speakers to interrupt each other. This paper describes a parser that can process input containing editing terms, speech repairs, and second speaker interruptions, and include these phenomena in its output. Such a parser allows a dialog system to reason about why editing terms were uttered; maybe the speaker was uncertain, embarrassed, reluctant to commit, etc. The reparandum (corrected material in a speech repair) also plays an important role as it may be referenced later: *take the oranges to Elmira uh I mean take them to Corning*. Reparanda may also give insight into the speaker's intentions: *pick up tankers in uh how many cars can an engine pull?*. Second speaker interruptions can provide evidence that the interrupter is listening (if they utter a backchannel such as *uh-huh*) or that neither speaker is hearing the other (both speakers are talking at the same time). This type of evidence is crucial for applications such as business meeting summarization.

Dialog systems are in their infancy. Systems use modules such as speech recognizers, parsers, reasoning systems, text generators, and speech synthesizers to interact with users. Researchers are just starting to experiment with better interfaces between these modules to improve performance. For example, some speech recognizers give parsers the n-best word sequences they find instead of just the sequence they assign the highest probability. Another area of development involves going beyond the typical command-response interface of a dialog system and allowing users to interrupt the system and speak more than one utterance per turn.

Clearly humans exhibit both of these behaviors when talking to each other. It is also uncontroversial that humans have a high degree of communication between processes in the brain that decode words, recognize syntactic structure, and perform general reasoning. Lower level information such as word stress is something that we can reason about and pragmatic expectations of what someone is likely to say can help word recognition.

Our work has focused on creating a parser that can process a stream of words with editing terms (*um*, *I mean*), speech repairs, and second speaker interruptions. This parser provides a syntactic representation to higher-level reasoning modules (such as a dialog manager) that includes this information. This parser is novel in that parsers typically make the simplifying assumption that any editing terms or speech repairs will be removed from the input. Another aspect neglected by current parsers is that people interrupt each other in conversation. Our parser allows second speaker interruptions and continuations. Thus, it can handle third party human-human conversations as well as allowing users to interrupt the system and vice versa.

In the first section of the paper, we describe why editing terms, speech repairs, and second speaker interruptions are important pieces of information that a parser needs to accommodate. The second section details how our parser handles these phenomena (described in more detail in (Core & Schubert 1998)) and the third section investigates assumptions made by the parser.

The corpus of data used in this work is the TRAINS-93 dialogs (Heeman & Allen 1995), a collection of human-human problem solving dialogs in a railway transportation domain. In these two speaker dialogs, one speaker is given a set of delivery goals[1] to achieve; the other speaker acts as a problem solving assistant, responsible for carrying out the plan. Examples from the TRAINS-93 domain will be used throughout the paper.

Motivation

A speech repair corrector can remove the majority of editing terms and speech repairs from the parser's input (e.g., on the basis of prosodic cues and local word patterns). Such an approach prevents the parser from using its grammatical information to improve speech repair identification (Core & Schubert 1999) and prevents reasoning about the meaning of hesitations and

[1] The cities in this domain are Avon, Bath, Corning, Dansville, and Elmira.

repairs in the input. Speech repairs are not always disruptive to human sentence processing (Fox Tree 1995) and often contain information about the speaker's mental state (for example, they may indicate uncertainity or embarrassment over part of the utterance).

One of our initial motivations for keeping editing terms and speech repairs in the input was that their content was important to the dialog manager. For example, a speaker could make reference to the reparandum (corrected material) of the speech repair: *take the oranges to Elmira uh I mean take them to Corning*. The dialog manager could also analyze the reparandum to get information about the speaker's plans. For example if the user uttered: *and pick up a tanker in wait how many cars can an engine carry?* an ideal system might respond *three cars, where do you want to pick up the tanker?* or it could even try to guess which tanker, *three cars, do you want to pick up the Avon tanker?*.

(Smith & Clark 1993) gives evidence that "fillers" and "hedges" are correlated with incorrect answers in question answering experiments. They define fillers as cases where speakers "used interjections such as *uh* and *oh*, and sometimes sighed, whistled, or talked to themselves" (p. 34). This definition evidently covers many editing terms, and in fact they say that *uh* and *um* are the most frequent fillers. (However, they do not analyze them separately.) Hedges are brief word sequences expressing uncertainty, such as *I guess*. Though infrequent, these were also associated with incorrect answers. The most frequently observed hedges (*I guess*, 11 examples and *I think*, 4 examples) are again editing terms.[2]

(Brennan & Williams 1995) extends these experiments to see if listeners perceive answers containing fillers as more likely to be wrong. They recorded answers to general knowledge questions and spliced together examples varying in the presence of *um* and *uh* as well as pauses and rising intonation. Subjects listened to these example answers (but not the preceding questions) and judged whether the answer was correct. Subjects marked answers less likely to be correct when following *um* or *uh* than when following a pause of the same length. Since this study also showed that length of pause correlates with perceived uncertainty, it follows that insertion of *um* and *uh* causes listeners to doubt the speaker's answer. A dialog system could mirror human reaction and assign different degrees of belief to user utterances depending on the presence of editing terms such as *um* and *uh*. More specifically it may be the case that phrases immediately after the filler should be viewed as suspect. In the example, *take the tankers to um Avon*, the system might question whether the NP, *Avon* is really the correct location. (Brennan & Schober 1999) notes that when dialog systems detect uncertainty they can offer clarifications such as in the hypothetical exchange below:

```
S: attach the aftercooler to the pump
U: um okay
S: the aftercooler should be the object to the
     right of the pump
```

(Brennan & Williams 1995) also points out that hesitation can result from reluctance to answer. This can happen when an answer is embarrassing, potentially getting the speaker into trouble (A: *I need my text book* B: *um I can't find it*), or makes an unpleasant commitment (A: *can you drive me to the airport at 2 am* B: *well okay*). Note that this can be crucial information if the answer is not entirely honest.

(Fox Tree 1999) also looked at the role of *um*s in answers to questions. In this case, the questions queried the listener's opinion (*are you here because of affirmative action?*). Recordings of these questions and their answers were digitally altered so that the interval between the question and answer contained pauses of various lengths and sometimes the word *um*. A set of subjects listened to these questions and answers and rated (1) how well the speakers knew each other, (2) whether the second speaker would seek further contact with the questioner, (3) how much speech production difficulty the second speaker had, (4) how deceptive the second speaker was being, and (5) how comfortable the second speaker was with the question topic. For speech production difficulty and comfort with topic, *um*s and long pauses both contributed negatively to listener interpretations and having both was worse than either alone. For familiarity and honesty judgments, an *um* or a pause contributed negatively but having both did not worsen judgments.

In addition to signaling uncertainty, lack of familiarity, dishonesty, anxiety, and speech production difficulty, fillers play an active role in the conversation by signaling that the speaker is trying to form an utterance but is having trouble. (Bortfeld *et al.* 1999) discusses this turn-taking function of fillers, and notes that after a filler is uttered, listeners can either give the speaker time to finish the utterance or jump in and suggest continuations. Bortfeld *et al.* cite studies showing that disfluencies (speech repairs and editing terms) in general decrease when there is no turn-taking (a monolog) and increase when visual turn taking cues cannot be used, as in telephone conversations.

(Brennan & Schober 1999) reports on a study showing that speech repairs and editing terms indicate planning difficulty. In the study, one speaker is the problem solver and the other acts as an assistant. When a speaker switches roles from assistant to problem solver, they generally have a higher rate of speech repairs and editing terms. Switching to a more difficult task also increases these rates. A dialog system might try to grab more initiative in these cases in order to help the user.

Most parsers, besides expecting a "sanitized" version of the input from which editing terms and repairs have been deleted, also expect to receive only one utterance at a time, uttered by a single speaker. The reason for

[2] The hedge, *something* also occurs 4 times in this corpus. It is unclear whether it counts as an editing term.

disallowing second speaker interruptions is to prevent the production of overlapping utterances. But this also rules out cooperative sentence formation, where one speaker continues or completes another speaker's utterance. Our goal has been to design a parser that could be used in a dialog system allowing multiple utterances per turn as well as interruptions by the user or the system. Such an ability would make the interaction with a user more natural. It would also be essential for non-interactive processing of free-flowing human-human dialogs, for instance for meeting summarization.

If a dialog manager is given information about second speaker interruptions, it can better track the grounding (Clark & Schaefer 1989) of the dialog. A dialog manager needs to know where backchannels such as *mm-hm* occur. If they occur in the middle of a sentence then they do not ground the whole sentence. Words spoken when both speakers are talking at the same time need to be marked as potentially not heard. When one speaker continues another speaker's utterance, this contribution grounds the initial utterance or utterance fragment; the original speaker must then accept the continuation if it is to enter the common ground.

To summarize, a parser that accommodates and analyzes editing terms, speech repairs, and second speaker interruptions, allows:

- improvement of speech repair identification using the parser's grammatical knowledge
- use of editing terms and repairs as indicators of uncertainty and processing difficulty
- processing of reparanda
  - in case their contents are later referenced
  - to determine the speaker's initial thoughts
- continuation of one speaker's utterance by another speaker
- tracking the grounding of the conversation

We next describe how the parser accommodates editing terms, speech repairs, and second speaker interruptions.

The Dialog Parser

The parser accepts a word stream that can contain words by either speaker in a two person dialog as well as editing terms and speech repairs. Editing terms are considered separate utterances located within other utterances. Speech repairs form parallel utterances; the corrected utterance skips the reparandum while a possibly incomplete utterance includes the reparandum. At each change of speaker, the new word by the second speaker may either 1) continue the first speaker's utterance, 2) continue a previous utterance by the second speaker, or 3) start a new utterance.

To implement this representation of dialog no changes are made to the parser's grammar itself. Instead metarules operate on the grammar to specify allowable

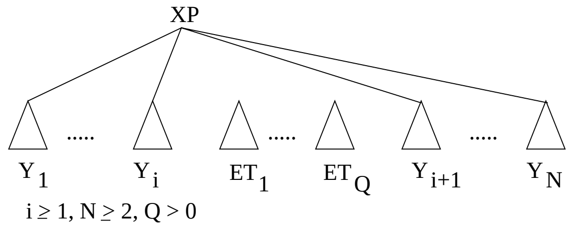

Figure 1: The Editing Term Metarule

patterns of phrase breakage and interleaving. Note, unlike traditional grammatical metarules, our metarules do not generate new grammar rules. Rather, they tell the parser how utterances may be interleaved and what partial utterances may be part of the dialog analysis. The *editing term metarule*, *repair metarule*, and *non-interference metarule* handle editing terms, speech repairs, and second speaker interruptions respectively.

The editing term metarule is shown in figure 1; sequences of one or more editing terms may appear between subconstituents of a phrase. The metarule allows the parser to extend copies of phrase hypotheses over editing terms that follow (where possible editing terms are defined in the lexicon). Copies are extended because the editing term may be a word such as *okay* that has other meanings that can be incorporated into the current phrases being constructed.

The repair metarule is shown in figure 2 and a sample of its application is shown in figure 3. The metarule extends copies of phrase hypotheses over reparanda indicated by a speech repair identifier. The alteration (Z'_1 through Z'_U) refers to words following the interruption point that correspond to words in the reparandum. These words are either repetitions or replacements (share the same part of speech). Many times the words Y_1 through Z_L will not form a complete utterance. Using statistical techniques to rank the set of partial utterance hypotheses ending at Z_L, the parser can narrow the alternatives and mark a small set of them as an analysis of what the speaker started to say.[3] The parser performs this disambiguation step for the corrected utterance as well. Note that we impose no restriction on who utters the correction so it may be a self-repair or a second-speaker correction.

It may be the case that the only utterance constituents covering the input do not skip a proposed reparandum. In that case, the parser can rule the reparandum a false alarm.[4] The connection between

[3]In some cases, the parser will have to gather pieces such as an NP and an incomplete VP to build this analysis.

[4]False alarm detection is actually more complicated than this because we need to compare probabilities of syntactic analyses covering the input (ones that include the reparandum and ones that do not) before deciding whether a false alarm has occurred. Note that the probabilities of repairs are included in the probabilities of analyses that skip the

the parser and the speech repair identifier should be flexible enough so that if the parser cannot find a syntactic analysis covering the input, it can query the speech repair identifier for other possible repairs in the input. One way to implement such an approach would be to give the parser a ranked list of possible sets of repairs in the utterance.[5]

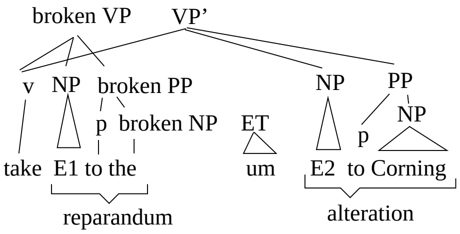

Figure 3: Sample Repair

The non-interference metarule is shown in figure 4. Copies of phrase hypotheses ending at Y_i may be extended over an interruption by another speaker (Z_1 through Z_Q). If an utterance includes two or more second speaker contributions, there are limitations on the phrase hypotheses allowed to be continued. Consider the following example:

```
AAA        AAA
    BBBB       BBBB
   1     2    3
```

Speaker A makes a contribution, followed by speaker B, then A again, and lastly B. At change of speaker 2, copies of phrase hypotheses ending at 1 (call these hypotheses α) are extended to 2 allowing B's contribution to be considered separately. At change of speaker 3, phrase hypotheses ending at 2 are extended to 3 except for those in α. B may continue his or her previous contribution or continue A's contribution ending at 3 but not A's contribution ending at 1.

Questioning the Framework

One way of testing this framework is to examine a corpus to see whether the proposed metarules deal adequately with the observed instances of editing terms, speech repairs, and second-speaker interruptions. We have carried out such an analysis for a set of transcriptions from the TRAINS-93 dialogs (Heeman & Allen 1995). In order to apply the metarules in the form outlined above, the contributions of the two speakers need to be merged into a single linear word stream (naturally, with each word annotated with the speaker identity). To accomplish this linearization, if one word overlaps another in time, then the word that ends earlier is placed in the parser's word stream prior to the other.

We examined 31 TRAINS dialogs[6] containing 3441 utterances,[7], 19029 words, 258 examples of interleaved utterances, and 481 speech repairs. All of the examples turned out to conform with our metarules.[8]

However, one may wonder whether important information about word overlap is lost in the single-streaming of the two speakers' contributions. If that were the case, then a model based on separate streams for the two participants would be preferable. It turns out that among the 258 examples of interleaved utterances, only three lead to word orderings that are potentially problematic. One example involves utterances 3 and 4 from d92a-4.3:

```
3.1 s: the five boxcars of oranges
3.2     that  are
4   u:      at   at Corning
```

Here, the user interrupts too early but repeats himself allowing a correct continuation. Another example involves utterances 60 and 61 from d92a-4.4:[9]

```
60 s: engine two is  loading
61 u:                is       is loading
```

Here, *u* intends to complete *s*'s utterance but is late in interrupting. This would be analyzed in a fashion similar to a repetition by a single speaker although in this case *u* grounds utterance 60 through the repetition.

Utterances 80 and 81 from d92a-1.2 are trickier to analyze and we show the positions of the actual word endings beneath the corresponding words:

```
80 u:   so       the        total     is
81 s:                    five
     255.5   255.56 255.83  256 256.61
```

Listening to the speech shows that *five* starts and ends in the middle of *total*. This is a case of the listener anticipating the question and giving an early answer. *s*

reparanda of these repairs. So all factors being equal, an analysis including a repair is less likely.

[5] A dialog manager can interact with our parser in a similar manner because the parser provides a ranked list of syntactic analyses instead of just the one its ranks highest.

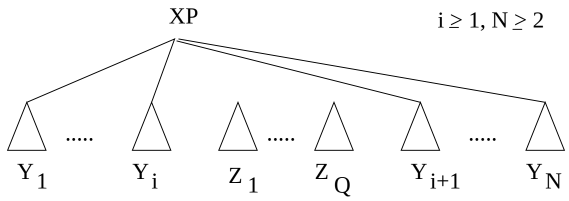

Figure 4: The Non-interference metarule

[6] Specifically, the dialogs were d92-1 through d92a-5.2 and d93-10.1 through d93-14.1

[7] This figure does not count editing term utterances nor utterances started in the middle of another speaker's utterance.

[8] This is not to say that a parser would actually find the desired analyses, since this of course depends on grammatical and lexical coverage, on reliable identification of possible reparandum and sentence boundaries, and correct disambiguation.

[9] Utterance 60 is abbreviated here for space reasons; its full form is: *goes on to Bath um while engine two is loading.*

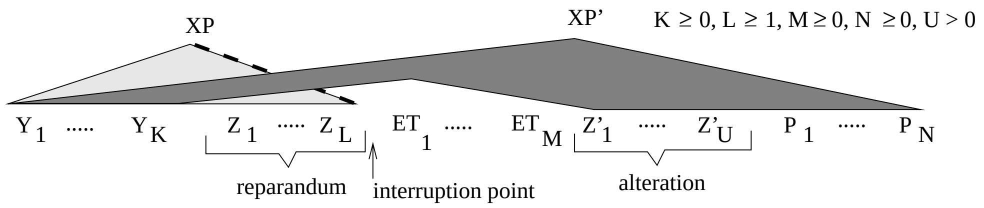

Figure 2: The Repair Metarule

repeats himself, giving evidence that he was not sure that he was understood:

```
82 s: that is right
   s: okay
83 u: five
84 s: so total is five
```

Though this example presents a problem for the dialog manager (which needs to make sense of the parsed utterances), the problem is not solved by paying closer attention to the relative temporal positions of *total* and *five*. Even if *five* is recognized as being uttered during *total*, the fact remains that the answer to the question precedes the completion of the question; the dialog manager will simply have to allow for such occurrences.

Exact word positions become important when one speaker continues another's utterance. It is difficult to determine word order given that the words of two speakers may overlap and that there is a delay between when a speaker decides to interrupt and the actual physical interruption. One could complicate the parsing model by removing the assumption that words are disjoint. Words could be given start and end indices and the parser could try various orderings of overlapping words. Currently there is no compelling evidence for such a complicated architecture. More investigation into overlapping speech is necessary but in the current data, our simple method of aligning words does as well as any more sophisticated technique.

The use of one word stream for two speakers has a pragmatic argument in its favor: it requires fewer modifications to a conventional parser. It requires implementation of only the non-interference metarule. A two stream parser involves a second set of parser data structures, a mechanism for switching between them, and implementation of a metarule that can link the word streams.

A last point to consider is that although the dialog manager may store a representation of the entire dialog seen so far, it is nearly impossible to reference the reparandum of a speech repair more than a few utterances away. One solution would be to implement some memory decay causing the repairs to be deleted if not referenced after a couple utterances. A more straightforward approach would be to modify search strategies to not look at repairs beyond a certain point in the past.

Conclusions

This paper presents a syntactic representation that allows more cooperation between higher-level reasoning modules and modules dealing with the speech signal. A major next step will be adding prosody to the representation starting with silence and word stress and eventually including intonation patterns (such as questioning tones or tones of surprise). The current representation allows the system to reason about editing terms, speech repairs, and second speaker interruptions. Editing terms can be taken as hesitations caused by uncertainty, processing difficulty, embarrassment, or reluctance. Reparanda of speech repairs are available for later reference in the dialog or for processing to better determine user intentions.

This representation also allows a better interface between the user and system as well as allowing the system to follow human-human dialog. Second speakers are allowed to interrupt or continue first speaker utterances. When speakers talk at the same time, this shows up as a series of second speaker interruptions. A dialog manager could take this as a signal that neither message was totally understood and that the second speaker was either expressing disagreement or had urgent information to convey.

Speech recognition errors are the most obvious limitation of current dialog systems. Dialog systems acting as problem-solving assistants also suffer from significant delays caused by their planners. However, it would be short-sighted not to assume that these technologies will improve to the point where current assumptions about editing terms, speech repairs, and turn-taking noticeably hinder the system. Even with current levels of speech recognition accuracy, editing terms and speech repairs are useful indicators of speaker uncertainty. These indicators might be used to hedge the information provided by user inputs: the dialog manager may interpret *there are oranges at um Avon* as if the user had said *maybe oranges are at Avon*. Also, when a user indicates uncertainty, a dialog system could take more initiative and provide clarification or planning assistance.

Acknowledgments

This work was supported in part by National Science Foundation grants IRI-9503312 and 5-28789. Thanks to James Allen, Peter Heeman, and Amon Seagull for their help and comments on this work.

References

Bortfeld, H.; Leon, S. D.; Bloom, J. E.; Schober, M. F.; and Brennan, S. E. 1999. Which speakers are most disfluent in conversation, and when? In *Notes of the ICPhS Satellite Meeting on Disfluency in Spontaneous Speech*, 7–10.

Brennan, S., and Schober, M. 1999. Speech disfluencies in spoken language systems: A dialog-centered approach. In *NSF Human Computer Interaction Grantees' Workshop (HCIGW 99)*.

Brennan, S. E., and Williams, M. 1995. The feeling of another's knowing: Prosody and filled pauses as cues to listeners about the metacognitive states of speakers. *Journal of Memory and Language* 34:383–398.

Clark, H. H., and Schaefer, E. F. 1989. Contributing to discourse. *Cognitive Science* 13:259–294.

Core, M., and Schubert, L. 1998. Implementing parser metarules that handle speech repairs and other disruptions. In Cook, D., ed., *Proc. of the 11th International FLAIRS Conference*.

Core, M., and Schubert, L. 1999. Speech repairs: A parsing perspective. In *Notes of the ICPhS Satellite Meeting on Disfluency in Spontaneous Speech*, 47–50.

Fox Tree, J. E. 1995. The effects of false starts and repetitions on the processing of subsequent words in spontaneous speech. *Journal of Memory and Language* 34:709–738.

Fox Tree, J. E. 1999. Between-turn pauses and ums. In *Notes of the ICPhS Satellite Meeting on Disfluency in Spontaneous Speech*, 15–17.

Heeman, P., and Allen, J. 1995. the TRAINS 93 dialogues. TRAINS Technical Note 94-2, Department of Computer Science, University of Rochester, Rochester, NY 14627-0226.

Smith, V. L., and Clark, H. H. 1993. On the course of answering questions. *Journal of Memory and Language* 32:25–38.