Implementing Parser Metarules that Handle 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

Mixed-initiative dialogs often contain interruptions in phrase structure such as repairs and backchannel responses. Phrase structure as traditionally defined does not accommodate such phenomena, so it is not surprising that phrase structure parsers are ill-equipped to handle them. This paper presents metarules that specify how the instantiations of phrase structure rules may be restarted or interrupted, with allowance for interleaved speech. In the case of interleaved speech or backchannel responses, the metarules allow syntactically separate constituents to interleave or to straddle each other. In the case of repairs, the metarules operate on the reparandum (what is being repaired) and alteration (the correction) to build parallel phrase structure trees: one with the reparandum and one with the alteration. Consider the partial utterance, *take the ban- um the oranges.* The repair metarule would build two VPs, one being *take the ban-* and the other being *take the oranges.* The introduction of metarules simplifies the notion of an *utterance* since a sentence interrupted by an acknowledgment such as *okay* can still be treated as a single utterance formed around the interrupting acknowledgment. Together, metarules and phrase structure rules specify the structures that should be accommodated by a parser for mixed initiative dialogs. A dialog parser should also maintain a dialog chart that stores the results of syntactic and semantic analysis of all the dialog seen so far. This dialog chart will be a shared resource eliminating the need for maintenance of a separate representation of dialog structure by a dialog manager.

Motivation

Repairs and other interruptions of phrase structure (such as backchannel acknowledgments) are common in mixed-initiative dialogs. Whether a dialog system is actively participating in such dialog or just following along, it needs to be able to correctly interpret repairs and interleaved speech.

This paper presents metarules that license certain extra-grammatical, interleaved structures. The first version of these metarules was presented in (Core & Schubert 1997); however, experience working with the rules has produced the revisions presented here.

Metarules either allow the parser to skip over certain material (parsing this material separately) or in the case of repairs, construct alternative representations (one for the erroneous material and one for the correction). Treating repairs and other disruptions of phrase structure as extra-grammatical simplifies the definition of "utterance" since interruptions do not necessarily end an utterance. We will use the term "utterance" to refer informally to a sentence or a single phrase (question answer) uttered by a speaker, allowing for repairs and acknowledgments.

Traditionally, dialog systems have only accepted one uninterrupted utterance at a time. The parsers of these systems send an utterance's logical form to some kind of dialog manager, a higher-level reasoning module that guides the system. There has been little if any discussion on how this logical form would encode any repairs in the utterance and how it could be extended to handle interleaved utterances. Dialog parsing metarules allow the chart of a dialog parser to contain parallel syntactic structures (what was first said and its correction) in the case of repairs, and interleaved syntactic structures in the case of interruptions. Instead of sending off this chart after an utterance is processed, it makes sense to treat the dialog chart as a shared resource between the dialog manager and parser in order to save communication costs. After each utterance, the chart would not be erased but rather appended, creating a record of the dialog for the dialog manager. The dialog chart could also serve as a place for the parser to mark interruptions such as *okay* as possible acknowledgments, and with the help of syntactic clues, mark interrogative, imperative, and declarative utterances as possible questions, suggestions, requests, and informs. For more details

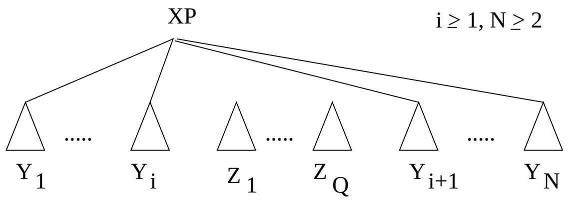

Figure 1: Skipping over an interruption by another speaker

see (Core & Schubert 1997).

The next sections concern dialog parsing in general and then focus on determining when to apply the repair metarule.

Dialog Parsing

While metarules can allow interruptions and repairs to disrupt a stream of words, the parser must be directly modified to deal with multiple speakers. In this paper, we restrict ourselves to the two-speaker case where one speaker may be an artificial agent. A further assumption is that the words of the two word streams (one for each speaker) do not overlap; i.e., when one speaker begins or resumes speaking, the other immediately falls silent. Thus changes of turn are well-defined, and occur at word boundaries.

Since the words do not overlap, they can be fed directly into the parser, labeled with their speaker. For the most part, the chart parser processes successive words in the usual way; however, at changes of turn, copies of arcs (incomplete phrase hypotheses) ending at the *previous* change of turn are created and their indices are set so as to skip the intervening speech. This allows one speaker's constituents to skip over utterances by the second speaker. Note that since the original, unextended arcs remain, it is still possible for one speaker to continue a phrase of the other. The structure of a phrase formed around an interruption Z by another speaker is shown in figure 1. There are changes of turn between Y_i and Z_1 and between Z_Q and Y_{i+1}, and in general XP can have more than one interruption. Note that if Y_1-Z_Q formed a sentence, then once the end of the utterance was reached, the parser would realize that Y_1-Z_Q covers the input if Y_{i+1}-Y_N is considered an interruption.

How the parser "skips" material is important since the metarules to be described involve skipping over repairs and interruptions in the same way. The following example illustrates how the parser automatically skips intervening speech as just outlined.

```
u: the train                    leaves tomorrow
s:             leaves at noon
  0    1     2       3  4    5      6         7
```

```
 NP -> det. n from 0 to 1
 NP           from 0 to 2
 S -> NP . VP from 0 to 2
 VP           from 2 to 5
 S            from 0 to 5
* second change of turn seen *
 S -> NP . VP from 0 to 5
 VP           from 5 to 7
 S            from 0 to 7
```

To simplify the example, the VP arcs are omitted. Note that the parser does not store completed constituents such as (`NP -> det n`) as arcs. It just adds the NP to the chart. The first change of turn has no previous change of turn to which to refer so nothing happens. At the second change of turn, copies are made of all arcs ending at the previous change of turn and their indices set to end at the current change of turn. Observe that a complete analysis of a dialog segment can include multiple utterances that jointly cover all the words uttered, but the individual utterances need not be adjacent. In the example above, there are two complete sentential utterances (sharing the same subject NP) that end at points 5 and 7 respectively.

While the mechanism for skipping over words of another speaker is built into the parser, metarules are needed to handle extending arcs over repairs and self-interruptions such as *uh* and *I mean*.

Editing Term Metarule

The *editing term metarule* allows constituents to skip over words signaling turn keeping (*um*, *ah*) and repairs (*I mean*). Editing terms[1] are confined to a small set of words and short word sequences. The structure allowed by the editing term metarule is shown in figure 2. A constituent may be interrupted between two subconstituents by one or more editing terms, and a constituent can be interrupted in more than one location (not shown in figure). We defer discussion of how metarules are implemented in the parser to the Implementation section.

Repair Metarule

Editing terms suggest the presence of a repair; other clues include ungrammatical input and word repetition. Once the presence of a repair has been hypothesized, possible starting points and end points of the reparandum (the corrected material) must be hypothesized as well. The *repair metarule* then allows constituents to skip over the possible reparanda.

[1] The terms *editing term*, *reparandum*, and *alteration* are used here as in (Heeman 1997).

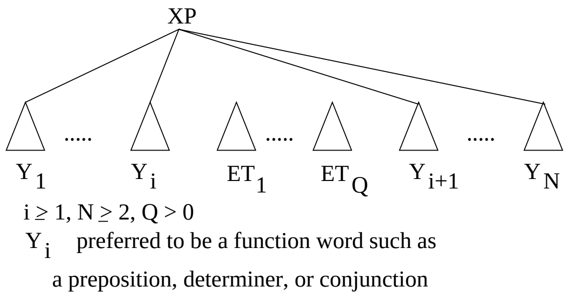

Figure 2: The Editing Term Metarule

Figure 3 is a picture of the general structure allowed by the repair metarule. XP is a possibly broken constituent ending with Z_L. It corresponds to what the speaker started to say before the correction. There may be more than one possible XP as people's speech is ambiguous especially when ended abruptly. One or more editing terms may indicate the end of the reparandum; these will be skipped via the editing term metarule. The corrected phrase, XP' includes the parts of the utterance before the reparandum, the alteration (the direct correction of the reparandum), and possibly additional words after the alteration. The words making up constituents, Y_i, Z_j, Z'_k, P_l, and ET_t, can be spoken by either of the speakers. Usually one speaker will correct himself/herself but there are cases where another speaker helps in the correction. Figure 4 shows syntax trees for an example repair: *take E1 to the um E2 to Corning.*

This definition provides no details about the inner structure of XP and XP' but with high probability the reparandum and alteration consist of a series of phrases (Z and Z') that are immediate subconstituents of XP and XP'. There is also a strong tendency for Z_i and Z'_i to be of the same type as well as for Levelt's rule (Levelt 1983) to hold: if Z is suitably completed then *Z and Z'* should form an allowable coordinate structure.

The reason the definition of the repair metarule is somewhat vague is that there are exceptions to the above tendencies. Consider this example from TRAINS dialog d93-8.2: *how does the long does that take.* Here, XP is a wh-question where part of the initial wh-phrase is repaired along with the auxiliary verb, the subject, and the VP.

Implementation

Determining when to apply the repair metarule is a subtle issue. Applying it too freely would bog down the parser, but we also do not want to miss actual repairs. Our initial goal is to determine to what extent we can improve repair detection and isolation by use of syntactic structures generated by the parser –

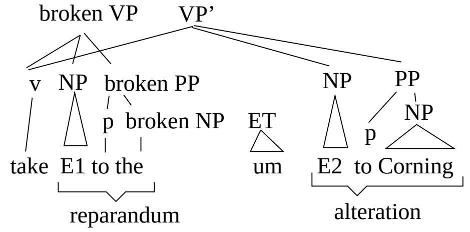

Figure 4: Sample Repair

which may be incomplete, inaccurate, and not fully disambiguated. We are building a corpus labeled with repairs to allow us to evaluate the adequacy of our approach (see below). A logical next step would be to measure parser accuracy with and without metarules; the metarules allow repairs and editing terms to be skipped, but do these end up in the most probable parse? This is a difficult question to answer for several reasons. We picked the TRAINS human-human dialogs because they have many repairs and interruptions. However, the language is not always grammatical and we do not have a grammar that fully covers the parts that are. The parser also does not yet have a mechanism for scoring parses probabilistically. (It uses a heuristic best-first approach.) As these obstacles are overcome, it will become possible to measure accuracy empirically based on at least some of the corpus. Interesting future work would be to see how the metarules help a partial parsing system create an interpretation when the regular parser fails.

How do the metarules interact? The editing term metarule operates on a constituent basis: if an editing term comes from the lexicon or is formed by a grammar rule (`et -> let's see`) then copies of all arcs ending in front of the editing term are set to end after the editing term.

The parser is incremental, processing one word at a time. After each word w_i is added to the chart, the editing term metarule may operate on the word if the word is, or completes, an editing term. Next, the probabilities of reparandum ending after words w_0 through w_{i-1} are updated based on the additional context provided by word w_i and the surrounding prosody. If one of the probabilities is over a certain threshold, then a corresponding repair beginning is determined. The repair metarule extends copies of arcs over this reparandum. Then repair ending probabilities have to be recomputed in case the new context (including a possibly repaired editing term) indicates another likely repair. Since the parser is not probabilistic, these probabilities are currently used solely to determine if the repair metarule should fire.

If a change of turn follows w_i, after the metarules

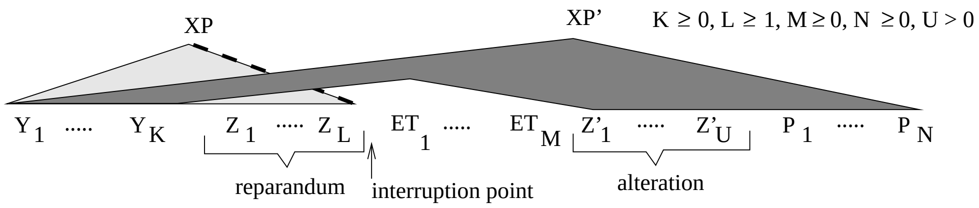

Figure 3: The Repair Metarule

have processed w_i, the parser extends copies of arcs from the previous change of turn to the end of w_i. The parser needs to be careful not to duplicate its efforts for interruptions composed solely of editing terms spoken by a different speaker. As stipulated here, a dialog parser would naturally extend copies of arcs past this interruption by another speaker but the editing term metarule would have already performed this work.

The following example illustrates the interactions between these metarules.

```
u: let's take engine E1   oh uh
s:                     E2?     to Corning right?
```

At the first change of turn, there is no previous change of turn so no arcs are extended. At the second change of turn, copies are made of all arcs ending after *E1* and are set to end after *E2*. The repair metarule recomputes its probabilities, and let's say that it selects *E1* as a reparandum. Copies of all the arcs before *E1* are extended to end after it. The parser is restarted and presumably *E2* extends a few of these arcs. The parser is now ready to process *oh*. The editing term metarule fires and makes copies of all arcs ending before *oh* so that they end after it.[2] The same is done for *uh*. The arcs containing *let's take engine E1* and *let's take engine E2* are extended by *to Corning* to form two separate sentences. The final *right* might be interpreted as a separate utterance, assuming the grammar allows this. (Intuitively, this final utterance refers to the correction and extension made by speaker *s*.)

Previous Work on Repair Correction

The repair metarule could potentially create much ambiguity by positing repairs at every word, giving the parser the option of skipping each word in the input. The next two sections discuss previous efforts at correctly locating repairs as well as discussing how a dialog parser should locate repairs.

The problems of finding utterance boundaries, detecting reparandum end points, and finding reparandum beginnings are intertwined. However, in this paper, we are going to consider them separately in order to simplify the discussion.[3]

Some of the previous work in handling speech repairs assumed that a repair could be detected through some prosodic clue or syntactic disruption and concentrated on determining the repair's extent through similarities in what was repaired and its correction. Other research focused on repair detection since prosodic and grammatical clues by themselves are inadequate detectors of speech repairs.

Isolating the Repair

(Hindle 1983) used pattern matching and grammatical information to flag repairs once they were detected. The pattern matching was a preprocessing step to flag repeated segments of text (separated by an editing term such as *okay*) as repairs. Grammatical information was used during parsing to classify two constituents of the same syntactic category separated by an editing term as a repair. The point of this work was to use parallel structures to identify the start of the reparandum assuming that the end was marked by an editing term.

(Kikui & Morimoto 1994) also concentrated on finding a reparandum start. They used an algorithm originally designed for analyzing coordinate structures to find a reparandum and alteration with the most matching words/parts of speech. In addition, they tested the well-formedness of the utterances remaining after various potential reparanda were removed. These tests used a language model called an adjacency matrix that specified what lexical categories could follow one another.

Detecting Repairs

As pointed out in (Heeman 1997), neither editing terms, word fragments, nor prosodic clues by themselves are a reliable signal for speech repairs. Bear

[2]This includes the arc containing *let's take engine E1.*

[3]So even though the syntactic parallelism measures used in finding reparandum beginnings are relevant to finding reparandum end points, we do not discuss this here.

et al. (Bear, Dowding, & Shriberg 1992) used pattern matching to identify likely repairs, then ran their parser on the string around the potential repair with and without the reparandum removed in order to establish grammaticality based on whether these strings comprised phrases. This approach performed well on the 37 test cases but it remains to be seen how it would do on more data.

To identify repairs, Nakatani and Hirschberg (Nakatani & Hirschberg 1993) used a variety of cues such as the existence of an editing term or word fragment, word repetition, and changes in speech signal energy and fundamental frequency. They used a decision tree to combine evidence from these sources to achieve 83.4% recall and 93.9% precision in detecting and isolating speech repairs.

(Heeman 1997) describes a statistical model that assigns part of speech tags and identifies speech repairs based on the context of previous editing terms (*I mean*, *um*), boundary tones, pauses, and the previous words and their categories. A boundary tone is a prosodic feature at the end of an intonational phrase; an intonational phrase is a section of the speech signal with a pitch contour of a certain shape. To determine the extent of the repair, word/part of speech matching plays an important role. Knowing the probability of one word/part of speech following another, one can estimate the degree of grammaticality of the utterance if a certain reparandum is chosen. Using these approaches gives a recall of 76.79% and precision of 86.66% on repair detection and 65.85% recall and 74.32% precision on correction. These results appear low when compared to (Nakatani & Hirschberg 1993); however Nakatani and Hirschberg used test utterances having at least one repair while Heeman did not remove sentences with no speech repairs from the test set.

Finding Repairs and Utterance Boundaries

The currently most successful speech repair correctors, (Heeman 1997) and (Nakatani & Hirschberg 1993), operate as preprocessing routines before parsing. A dialog parser should enable better detection of syntactic disruptions that indicate repairs, since it has a record of the possible phrase structures in the input. In addition, it can compute a deeper syntactic parallelism between a potential reparandum and its alteration.[4]

[4]However, to aid speech recognition, preprocessing routines are needed to explain otherwise unlikely word co-occurrences. Assuming such a routine is available one could reformulate equation 1 as simply an update to a repair probability from a preprocessing routinue. Likewise, reparandum beginning probabilities would also be adjusted.

We currently have not implemented a reparandum ending detector nor an utterance ending detector. The plan is to use decision trees to calculate the utterance ending and reparandum ending probabilities presented below. Word matching is one way to compute a syntactic parallelism score. We are currently looking at ways to do better given the syntactic information available on the parser's chart.

To decide whether point x is an utterance boundary, the probability in expression (1) will be used, in comparison with the corresponding expression for $\neg B_x$.

$$P(B_x|B_{1,x-1}, T_x, DM_{x,t}, SS_{0,x}, COT_x)P(B_x|S_x)/P(B_x) \quad (1)$$

(For $\neg B_x$, each of the 3 probability terms $P(...)$ in (1) is replaced by $1 - P(...)$.)

- $B_{1,x-1}$ - utterance boundaries from 1 to x-1.
- T_x - any boundary tone at x.
- $DM_{x,t}$ - sequence of discourse markers following x.
- $SS_{0,x}$ - syntactic structure from 0 to x.
- COT_x - any speaker change at x.
- S_x - length of silence at x.

The amount of silence training data in the TRAINS dialogs is small, so silence is treated as a separate evidence item in 1 as suggested in (Heeman 1997). Technically, the assumption is that $B_{1,x-1}, T_x, DM_{x,t}, SS_{0,x}, COT_x$ are independent of silence length S_x, given B_x. Let CT abbreviate $B_{1,x-1}, T_x, DM_{x,t}, SS_{0,x}, COT_x$. The stated assumption gives the approximation

$$P(B_x|CT, S_x) \approx \frac{P(B_x|S_x)P(B_x|CT)P(CT)}{P(CT|S_x)P(B_x)} \quad (2)$$

However, as indicated we are interested in comparing $P(B_x|CT, S_x)$ with $P(\neg B_x|CT, S_x)$. The expressions for these probabilities both have the factor $P(CT)/P(CT|S_x)$ so it can be left out of (1).

To use syntactic structure as context, we plan to use an extremely simple model that tests whether position x is a clause or phrase ending and uses this as additional evidence.

To detect repair endings, we need to compute the probability of a fresh start ending at a certain position j and the probability of a modification repair ending at j. Fresh starts, repairs starting at the beginning of an utterance, are different in character from modification repairs (repairs not going back to the utterance start). As noted by Levelt, editing terms such as *sorry* occur more often with fresh starts and there may be other differences between the two types of repairs. An expression (analogous to (1)) for the probability of a modification repair $P(R_j)$ ending at j when the input up to point k has been seen is shown in equation (3); the fresh start probability is the same except that F_j (fresh start at j) replaces R_j. $ET_{j,q}$ is the presence of a sequence of editing terms starting at j and ending at q with $j < q < k$.

$$P(R_j|ET_{j,q}, T_j, SS_{0,k})P(R_j|S_j)/P(R_j) \qquad (3)$$

The best way to use syntactic structure to find reparandum endings is unclear. Each word has an associated list of constituents of which it is a part. Constituents such as sentences are more likely to signal grammaticality than sub-phrasal constituents. In addition, we need to consider whether these constituents extend to the left or right of the word, as it is less likely for a reparandum to end in the middle of a constituent.

Once a reparandum ending is located, a reparandum beginning must be found as well. Scores are assigned to possible reparandum beginnings based on the syntactic parallelism between the reparanda they define and the following alteration. We need to adjust the parallelism score for the reparandum starting at position 0 by the probability of a fresh start. Then the reparandum beginning with the highest parallelism score is chosen. The adjustment could be as simple as: $\alpha P(F_j) + (1-\alpha)score_0$ with α determined in training.

We are studying modification repairs to develop a good formula for syntactic parallelism. This work is still in progress as we decide how to filter the large numbers of constituents that the bottom-up parser puts on the chart. Usually in parsing, many of the constituents can be ignored because they do not contribute to a full parse. In this case, we are not going to have a correct full parse because of the repair.

Conclusions

Instead of trying to hide repairs and other interruptions from the parser, the proposed framework allows the parser to accommodate them and not leave out important aspects of the dialog structure when presenting the dialog manager with a representation of the dialog. In addition, this framework enables a tighter coupling between the dialog manager and parser through a shared dialog chart.

Another reason for having the parser accommodate repairs, hesitations, and interruptions is that the parser has information about the syntactic structure of the utterance and what are allowable structures. These are sources of information that preprocessing routines do not have; the dialog parser can still use acoustic cues, pattern matching, and other sources of information that preprocessing techniques use.

In addition to the promise of a parser specially tailored to processing dialogs, this work contributes a formal description of the structure of mixed initiative-dialog - with all its interruptions, interleaving, and repairs. This description finesses the problem of what an "utterance" is by using traditional phrase structure as our theory of grammatical structure and using metarules to model the disruption of this structure in dialogs.

Acknowledgments

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

References

Bear, J.; Dowding, J.; and Shriberg, E. 1992. Integrating multiple knowledge sources for detection and correction of repairs in human-computer dialog. In *Proc. of the 30th annual meeting of the Association for Computational Linguistics (ACL-92)*, 56–63.

Core, M. G., and Schubert, L. K. 1997. Handling speech repairs and other disruptions through parser metarules. Technical Report SS-97-04, American Association for Artificial Intelligence (AAAI), 445 Burgess Drive, Menlo Park CA 94025.

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.

Heeman, P. A. 1997. *Speech Repairs, Intonational Boundaries and Discourse Markers: Modeling Speakers' Utterances in Spoken Dialog.* Ph.D. Dissertation, Department of Computer Science, University of Rochester.

Hindle, D. 1983. Deterministic parsing of syntactic non-fluencies. In *Proc. of the 21st annual meeting of the Association for Computational Linguistics (ACL-83)*, 123–128.

Kikui, G., and Morimoto, T. 1994. Similarity-based identification of repairs in Japanese spoken language. In *Proceedings of the 3rd International Conference on Spoken Language Processing (ICSLP-94)*, 915–918.

Levelt, W. J. M. 1983. Monitoring and self-repair in speech. *Cognitive Science* 14:41–104.

Nakatani, C., and Hirschberg, J. 1993. A speech-first model for repair detection and correction. In *Proc. of the 31st annual meeting of the Association for Computational Linguistics (ACL-93)*, 46–53.