# Improved Syntactic Models for Parsing Speech with Repairs[*]

**Tim Miller**
Department of Computer Science and Engineering
University of Minnesota, Twin Cities
tmill@cs.umn.edu

## Abstract

This paper introduces three new syntactic models for representing speech with repairs. These models are developed to test the intuition that the erroneous parts of speech repairs (reparanda) are not generated or recognized as such while occurring, but only after they have been corrected. Thus, they are designed to minimize the differences in grammar rule applications between fluent and disfluent speech containing similar structure. The three models considered in this paper are also designed to isolate the mechanism of impact, by systematically exploring different variables.

## 1 Introduction

Recent work in recognition of speech with repairs has shown that syntactic cues to speech repair can improve both overall parsing accuracy and detection of repaired sections (Hale et al., 2006; Miller and Schuler, 2008; Johnson and Charniak, 2004). These techniques work by explictly modeling the structure of speech repair, specifically the tendency of repairs to follow unfinished constituents of the same category. This is the essence of what was termed the *well-formedness rule* by Willem Levelt (1983) in his psycholinguistic studies of repair.

The work presented here uses the same motivations as those cited above (to be described in more detail below), in that it attempts to model the syntactic structure relating unfinished erroneous constituents to the repair of those constituents. However, this work attempts to improve on those models by focusing on the generative process used by a speaker in creating the repair. This is done first by eschewing any labels representing the presence of an erroneous constituent while processing the text. This modeling representation reflects the intuition that speakers do not intend to generate erroneous speech – they intend their speech to be fluent, or a correction to an error, and can stop very quickly when an error is noticed. This corresponds to Levelt's *Main Interruption Rule*, which states that a speaker will "Stop the flow of speech immediately upon detecting the occasion of repair." Rather than attempting to recognize a special syntactic category called EDITED during the processing phase, this work introduces the REPAIRED category to signal the *ending* of a repaired section only.

The second part of the modeling framework is the use of a *right-corner transform* on training data, which converts phrase-structure trees into heavily left-branching structures. This transformation has been shown to represent the structure of unfinished constituents like those seen in speech repair in a natural way, leading to improved detection of speech repair (Miller and Schuler, 2008).

Combining these two modeling techniques in a bottom-up parsing framework results in a parsing architecture that is a reasonable approximation to the sequential processing that must be done by the human speech processor when recognizing spoken language with repairs. This parser also recognizes sentences containing speech repair with better accuracy than the previous models on which it is based.

[*]This research was supported by NSF CAREER award 0447685. The views expressed are not necessarily endorsed by the sponsors.

*Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL*, pages 656–664,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

Therefore, these syntactic models hold promise for integration into systems for processing of streaming speech.

### 1.1 Speech Repair Terminology

A speech repair occurs when a speaker decides to interrupt the flow of speech and restart part or all of an utterance. Typically speech repair structure (Shriberg, 1994) is considered to contain a *reparandum*, or the part of the utterance to be replaced, and an *alteration*, which is meant to replace the reparandum section. There are also frequently *editing terms* (for example, 'uh' and 'um') between the reparandum and alteration, which may be used to signal the repair, or to indicate that the speaker is thinking, or just to maintain control of the dialogue.

### 1.2 Related Work

This work is related to that of Hale et al.(2006) in that it attempts to model the syntactic structure of speech repair. In that paper speech repair detection accuracy was increased by explicitly accounting for the relation between reparanda category and alteration category. This was done by so-called "daughter annotation," which expanded the set of EDITED categories by appending the category below the EDITED label to the end of the EDITED label – for example, a noun phrase (NP) reparanda would be of type EDITED-NP. In addition, this approach made edit detection easier by propagating the -UNF label attached to the rightmost unfinished constituent up to the EDITED label. These two changes in combination allow the parser to better recognize when a reparandum has occurred, and to make siblings of reparanda and alterations with the same basic category label.

Another model of speech repair that explicitly models the structure of speech repair is that of Johnson and Charniak (2004). That model has a different approach than the context-free parsing approach done in the present work. Instead, they run a tree-adjoining grammar (TAG) parser which traces the overlapping words and part-of-speech tags that occur in the reparandum and alteration of a speech repair. This approach is highly accurate at detecting speech repairs, and allows for downstream processing of cleaned up text to be largely free of speech repair, but due to its TAG component it may present difficulties incorporating into an architecture that operates on streaming text or speech.

This work is also similar in aim to a component of the parsing and language modeling work of Roark and Johnson (1999), which used right-binarization in order to delay decision-making about constituents as much as possible. For example, the rule

$$NP \rightarrow DT \;\; NN$$

might be right-binarized as two rules:

$$NP \rightarrow DT \;\; NP\text{-}DT$$

and

$$NP\text{-}DT \rightarrow NN$$

The result of this binarization is that when predicting the noun phrase (NP) rule, a top-down parser is delaying making any commitments about the category following the determiner (DT). This delay in prediction means that the parser does not need to make any predictions about whether the next word will be, e.g., a common noun (NN), plural noun (NNS), or proper noun (NNP), until it sees the actual next word.

Similarly, the model presented in this work aims to delay the decision to create a speech repair as much as possible. This is done here by eliminating the EDITED category (representing a reparandum) during processing, replacing it with a REPAIRED category which represents the alteration of a speech repair, and by eliminating implicit cues about repair happening before a decision to repair should be necessary.

Finally, this work is most directly related to that of Miller and Schuler (2008). In that work, the authors used a right-corner transform to turn standard phrase-structure trees into highly left-branching trees with sub-tree category labels representing incomplete but in-progress constituent structure. That structure was shown to have desirable properties in the representation of repair in syntax trees, and this work leverages that insight, while attempting to improve the input representation such that the right-corner representation does not require the parser to make any assumptions or decisions earlier than necessary.

## 2 Syntactic Model

This section will first describe the default representation scheme for speech repair in the Switchboard corpus and the standard representation after application of a right-corner transform, and then describe why there are shortcomings in both of these representations. Descriptions of several alternative models follow, with an explanation of how each of them is meant to address the shortcomings seen in previous representations. These models are then evaluated in Section 3.

### 2.1 Standard Repair Annotation

The standard representation of speech repair in the Switchboard corpus makes use of one new category label (EDITED), to represent a reparandum, and a new dash-tag (-UNF), representing the lowest unfinished constituent in a phrase. An example tree with both EDITED and -UNF tags is shown in Figure 1.

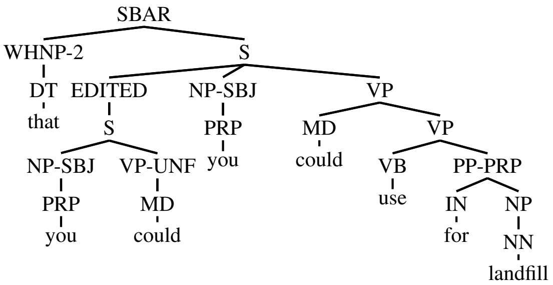


Figure 1: A fragment of a standard phrase-structure tree from the development set, containing both an EDITED constituent and an -UNF tag.

This sentence contains a restarted sentence (S) constituent, in which the speaker started by saying "you could", then decided to restart the phrase, in this case without changing the first two words. One important thing to notice is that the EDITED label contains no information about the structure beneath it. As a result, a parser trained on this default annotation has no information about the attempted constituent type, which, in the case of restarts would obviously be beneficial. As described above, the work by Hale et al. using daughter annotation was meant to overcome this shortcoming.

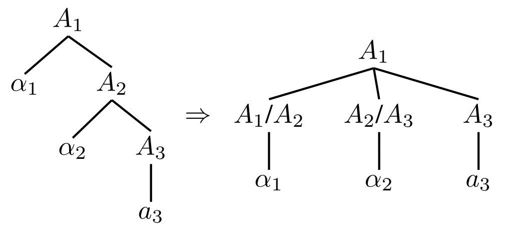
Another shortcoming of this annotation scheme to consider is that the EDITED tag is not meaningful with respect to constituent structure. Attempting to learn from this structure, for example a probabilistic context-free grammar, will result in the rule that a sentence (S) consists of a reparandum, a noun phrase, and a verb phrase, which is an odd way of thinking about both constituent structure and meaning. A more intuitive understanding might be that a sentence may consist of a noun phrase followed by a verb phrase, and during the production of that rule, an interruption may occur which causes the rule to restart.

### 2.2 Right-Corner Transform

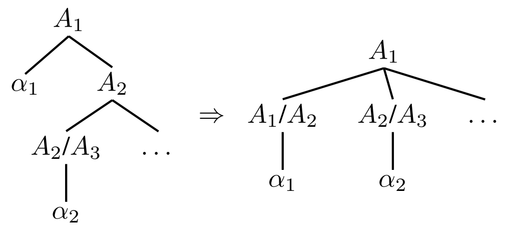
The work described above by Miller and Schuler (2008) uses a right-corner transform. This transform turns right-branching structure into left-branching structure, using category labels that use a "slash" notation $\alpha/\gamma$ to represent an incomplete constituent of type $\alpha$ "looking for" a constituent of type $\gamma$ in order to complete itself. Figure 2 shows the right-corner transformed tree from above.

This transform first requires that trees be binarized. This binarization is done in a similar way to Johnson (1998) and Klein and Manning (2003).

Rewrite rules for the right-corner transform are as follows, first flattening right-branching structure:[1]



(1)



(2)

then replacing it with left-branching structure:

[1] Here, all $A_i$ denote nonterminal symbols, and $\alpha_i$ denote subtrees ; the notation $A_1{:}\alpha_0$ indicates a subtree $\alpha_0$ with label $A_1$; and all rewrites are applied recursively, from leaves to root. In trees containing repairs, the symbol ET represents any number of editing terms and the sub-structure within them.

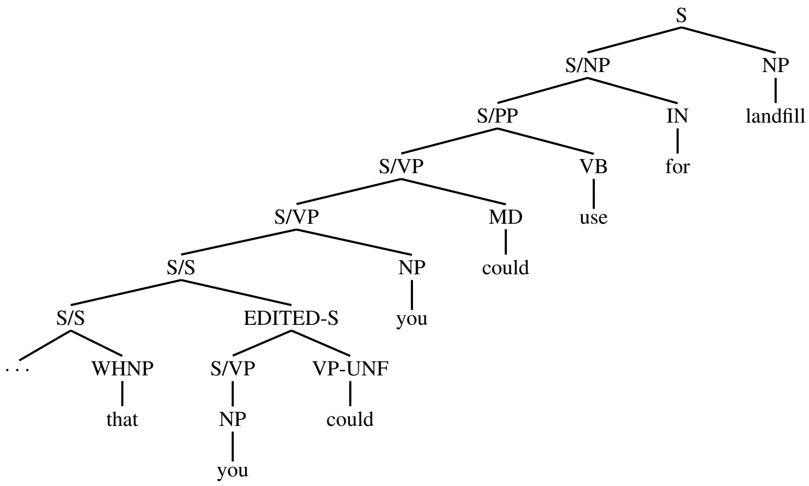
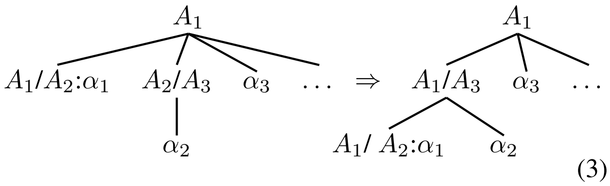


Figure 2: Right-corner transformed tree fragment.



(3)

This representation has interesting properties, which work well for speech repair. First, the left-branching structure of a repair results in reparanda that only require one special repair rule application, at the last word in the reparandum. Second, the explicit representation of incomplete constituents allows many reparanda to seamlessly integrate with the rest of the parse tree, with the EDITED label essentially acting as an instruction to the parser to maintain the current position in the unfinished constituent. This subtle second point is illustrated in the tree in Figure 2. After the EDITED section is detected, it combines with a category label S/S to form another sub-tree with category label S/S, essentially acting as a null op in a state machine looking to complete a phrase of type S.

This representation also contains problems, however. First, note that the (bottom-up) parser uses one set of rules to combine the reparandum with the current state of the recognition, and another set of rules when combining the alteration with the previous input. While it is a benefit of this approach that both rule sets are made up of fluent speech rules, their way of combining nonetheless requires an early premonition of the repair to occur. If anything, the repair should require special rule applications, but in this representation it is still the case that the reparandum looks different and the alteration looks "normal."

A better model of repair from a recognition perspective would recognize the reparandum as fluent, since they are recognized as such in real time, and then, when noticing the repeated words, declare these new words to be a repair section, and retroactively declare the original start of the phrase to be a reparandum. It is this conception of a recognition model that forms part of the basis for a new syntactic model of speech repair in Section 2.3.

A second problem with this representation is evident in certain multi-word repairs such as the one in Figure 2 that require an extra right branch off of the main left branching structure of the tree. As a result, a multi-word reparandum structure requires an extra unary rule application at the left-corner of the sub-tree, in this case S/VP, relative to the inline structure of the fluent version of that phrase. This extra rule will often be nearly deterministic, but in some cases it may not be, which would result essentially in a penalty for starting speech repairs. This may act to discourage short repairs and incentivize longer reparanda, across which the penalty would be amortized. This incentive is exactly backwards, since reparanda tend to be quite short.

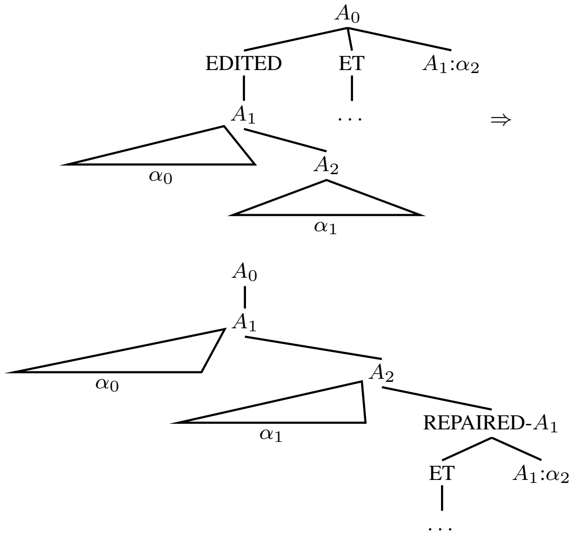
The next section will show how the two issues mentioned above can be resolved by making modifications to the original structure of trees containing repairs.

### 2.3 Modified Repair Annotation

The main model introduced in this paper works by turning the original repair into a right-branching structure as much as possible. As a result, the right-corner transformed representation has very flat structure, and, unlike the standard right-corner transformed representation described above, does not require a second level of depth in the tree with different rule applications. This can also be an important consideration for speech, since there are parsers that can operate in asymptotically linear time by using bounded stacks, and flat tree structure minimizes the amount of stack space required.

This model works by using an "interruption" model for the way a repair begins. The interruption model works on restarted constituents, by moving the repaired constituent (the alteration) to be the right-most child of the original EDITED constituent. The EDITED label is then removed, and a new REPAIRED label is added. This of course makes the detection of EDITED sections possible only retrospectively, by noting a REPAIRED section of a certain syntactic category, and tracing back in the tree to find the closest ancestor of the same category.

This can be illustrated schematically by the following rewrite rule:



(4)

Figure 3 shows how the example tree from Figure 1 looks when transformed in this manner. The result of these transformations may appear odd, but it is important to note that it is merely an intermediate stage between the "standard" representation with an EDITED label, representing the post-recognition understanding of the sentence, and the right-corner representation in which recognition actually occurs. This right-corner representation can be seen in Figure 2.3.

This representation is notable in that it looks exactly the same after the first word of the repair ('you') as the later incarnation of the same word in the alteration. After the second word ('could'), the repair is initiated, and here a repair rule is initiated. It should be noted, however, that strictly speaking the only reason the REPAIRED category needs to exist is to keep track of edits for the purpose of evaluating the parser. It serves only a processing purpose, telling the parser to reset what it is looking for in the incoming word stream.

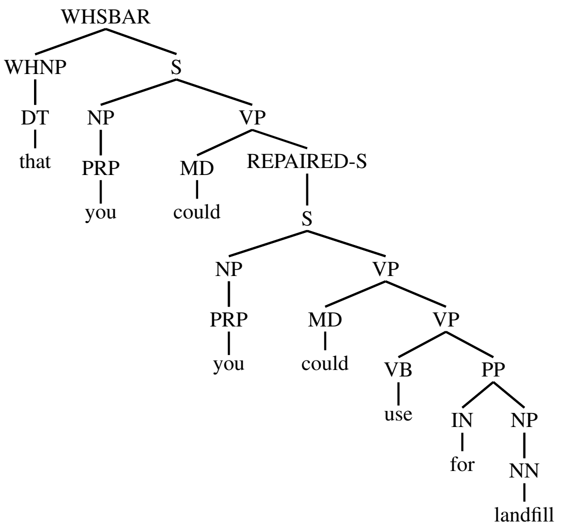


Figure 3: REPAIRED-INT transformation

The next model attempts to examine the impact of two different factors in the REPAIRED-INT representation above. That representation had the side effect of creating special rules off of the alteration (REPAIRED) node, and it is difficult to assign praise or blame to the performance results of that model without distinguishing the main modification from the side effects. This can be rectified by proposing another model that similarly eliminates the EDITED label for reparanda, and uses a new label REPAIRED for the alteration, but that

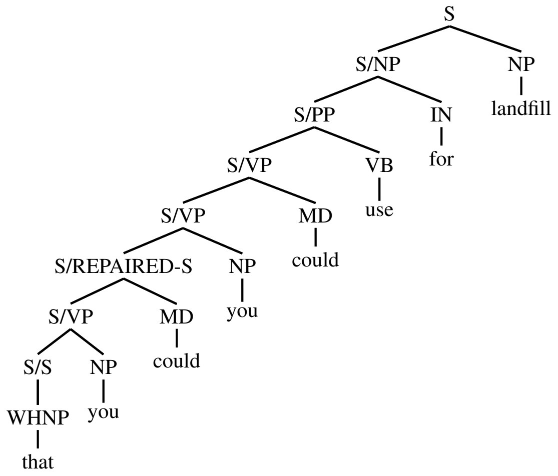
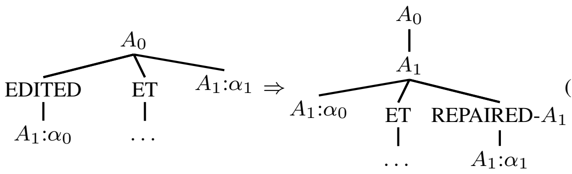


Figure 4: REPAIRED-INT + right-corner transformation

does not satisfy the desire to have reparanda occur inline using the "normal" rule combinations. This model does, however, still have special rules that the REPAIRED label will generate. Thus, if this model performs equally well (or equally as poorly) as REPAIRED-INT, then it is likely due to the model picking up strong signals about an alteration rule set. This modification involves rewriting the original phrase structure tree as follows:



(5)

A tree with this annotation scheme can be seen in Figure 5, and its right-corner counterpart is shown in Figure 6.

The final modification to examine acts effectively as another control to the previous two annotation schemes. The two modifications above are essentially performing two operations, first acting to binarize speech repairs by lumping a category of type X with a category of type EDITED-X, and then explicitly marking the repair but not the reparandum. This modification tests whether simply adding an extra layer of structure can improve performance while retaining the standard speech repair annotation including the EDITED category label. This modification will be denoted EDITED-BIN.

EDITED-BIN trees are created using the following rewrite rule:

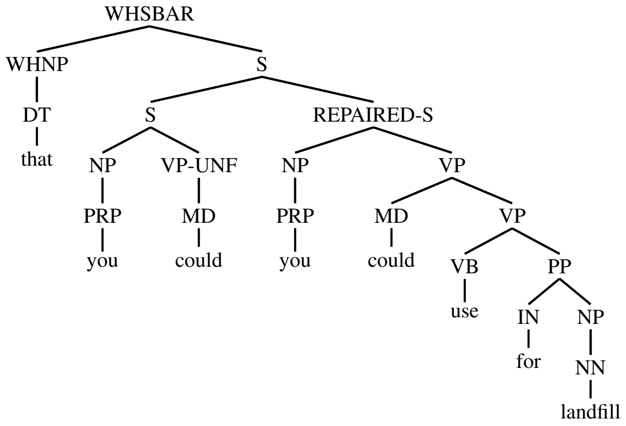


Figure 5: REPAIRED-BIN transformation

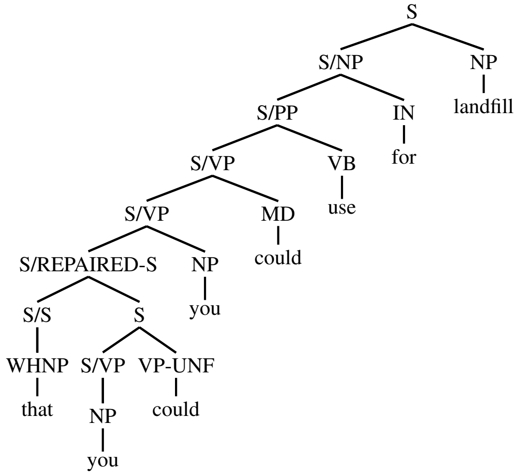
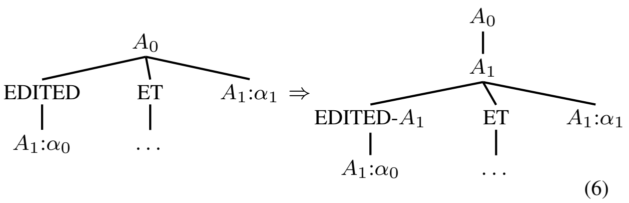


Figure 6: REPAIRED-BIN + right-corner transformation



(6)

After this transform, the tree would look identical to the REPAIRED-BIN tree in Figure 5, except the node labeled 'REPAIRED-S' is labeled 'S', and its left sibling is labeled 'EDITED-S' instead of 'S.' An EDITED-BIN tree after right-corner transformations is shown in Figure 7. This explicit binarization of speech repairs may be effective in its own right, because without it, a 'brute force' binarization must be done to format the tree before applying the right-corner transform, and that process involves joining chains of categories with underscores into right-branching super-categories. This process can result in reparanda categories in unpredictable places in the middle of lengthy super-categories, making data sparse and less reliable.

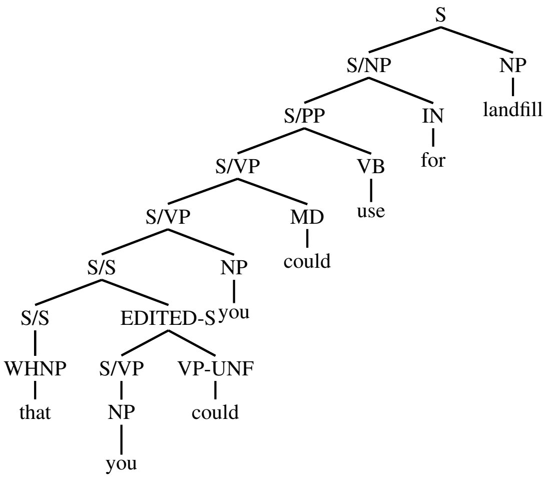


Figure 7: EDITED-BIN + right-corner transformation

## 3 Evaluation

The evaluation of this model was performed using a probabilistic CYK parser[2]. This parser operates in a bottom-up fashion, building up constituent structure from the words it is given as input. This parsing architecture is a good match for the structure generated by the right-corner transform because it does not need to consider any categories related to speech repair until the repaired section has been completed. Moreover, the structure of the trees means that the parser is also building up structure from left to right. That mode of operation is useful for any model which purports to be potentially extensible to speech recognition or to model the human speech processor. In contrast, top-down parsers require exhaustive searches, meaning that they need to explore interpretations containing disfluency, even in the absence of syntactic cues for its existence.

These experiments used the Switchboard corpus (Godfrey et al., 1992), a syntactically-annotated corpus of spontaneous dialogues between human interlocutors. This corpus is annotated for phrase structure in much the same way as the Penn Treebank Wall Street Journal corpus, with the addition of several speech-specific categories as described in Section 2.1. For training, trees in sections 2 and 3 of this corpus were transformed as described in Section 2, and rule probabilities were estimated in the usual way. For testing, trees in section 4, subsections 0 and 1, were used. Data from the tail end of section 4 (subsections 3 and 4) was used during development of this work.

Before doing any training or testing, all trees in the data set were stripped of punctuation, empty categories, typos, all categories representing repair structure, and partial words – anything that would be difficult or impossible to obtain reliably with a speech recognizer. A baseline parser was then trained and tested using the split described above, achieving standard results as seen in the table below. For a fair comparison to the evaluation in Hale et al. (2006), the parser was given part-of-speech tags along with each word as input. The structure obtained by the parser was then in the right-corner format. For standardized scoring, the right-corner transform, binarization, and augmented repair annotation were undone, so that comparison was done against the nearly pristine test corpus. Several test configurations were then evaluated, and compared to three baseline approaches.

The two metrics used here are the standard Parseval F-measure, and Edit-finding F. The first takes the F-score of labeled precision and recall of the nonterminals in a hypothesized tree relative to the gold standard tree. The second measure marks words in the gold standard as edited if they are dominated by a node labeled EDITED, and measures the F-score of the hypothesized edited words relative to the gold standard (recall in this case is percentage of actual edited words that were hypothesized as edited, and precision is percentage of hypothesized edited words that were actually edited).

The first three lines in the table refer to baseline approaches to compare against. “Plain” refers to a configuration with no modifications other than the removal of repair cues. The next result shown is a reproducton of the results from Hale et al. (2006) (described in section 1.2)[3]. The next line (“Standard

[2] The specific parser used is the Stanford parser described in Klein and Manning(2003), but run in “vanilla PCFG” mode.

[3] The present work compares to the standard CYK parsing result from that paper, and not the result from a heavily optimized parser using lexicalization.

Right Corner") is a reproduction of the results from Miller and Schuler (2008).

The following three lines contain the three experimental configurations. First, the configuration denoted EDITED-BIN refers to the simple binarized speech repair described in Section 2.3 (Equation 6). REPAIRED-BIN refers to the binarized speech repair in which the labels are basically reversed from EDITED-BIN (Equation 5). Finally, REPAIRED-INT refers to the speech repair type where the REPAIRED category may be a child of a non-identity category, representing an interruption of the outermost desired constituent (Equation 4).

| *System Configuration* | *Parseval-F* | *Edited-F* |
|---|---|---|
| Baseline | 71.03 | 17.9 |
| Hale et al. | $68.47^{\dagger\dagger}$ | $37.9^{\dagger\dagger}$ |
| Standard Right Corner | $71.21^{\dagger\dagger}$ | $30.6^{\dagger\dagger}$ |
| **EDITED-BIN** | $69.77^{**\ \dagger\dagger}$ | $38.9^{**\ \dagger\dagger}$ |
| **REPAIRED-BIN** | $71.37^{*}$ | $31.6^{**\ \dagger\dagger}$ |
| **REPAIRED-INT** | $71.77^{**}$ | $39.2^{**\ \dagger\dagger}$ |

Table 1: Table of parsing results. Star ($^*$) indicates significance relative to the 'Standard Right Corner' baseline ($p < 0.05$), dagger ($^\dagger$) indicates significance relative to the 'Baseline' labeled result ($p < 0.05$). Double star and dagger indicate highly significant results ($p < 0.001$).

Significance results were obtained by performing a two-tailed paired Student's t-test on both the Parseval-F and Edit-F per-sentence results. This methodology is not perfect, since it fails to account for the ease of recognition of very short sentences (which are common in a speech corpus like Switchboard), and thus slightly underweights performance on longer sentences. This is also the explanation for the odd effect where the 'REPAIRED-BIN' and 'REPAIRED-INT' results achieve significance over the 'Standard Right Corner' result, but not over the 'Baseline' result. However, the simplest alternative – weighting each sentence by its length – is probably worse, since it makes the distributions being compared in the t-test broadly distributed collections of unlike objects, and thus hard to interpret meaningfully.

These results show a statistically significant improvement over previous work in overall parsing accuracy, and obvious (as well as statistically significant) gains in accuracy recognizing edited words (reparanda) with a parser. The REPAIRED-INT approach, which makes repair structure even more highly left-branching than the standard right-corner transform, proved to be the most accurate approach. The superior performance according to the EDIT-F metric by REPAIRED-INT over REPAIRED-BIN suggests that the improvement of REPAIRED-INT over a baseline is not due simply to a new category. The EDITED-BIN approach, while lowering overall accuracy slightly, does almost as well on EDITED-F as REPAIRED-INT, despite having a very different representation of repair. This suggests that there are elements of repair that this modification recognizes that the others do not. This possibility will be explored in future work.

Another note of interest regards the recovery of reparanda in the REPAIRED-INT case. As mentioned in Section 2.3, the EDITED section can be found by tracing upwards in the tree from a REPAIRED node of a certain type, to find an non-repaired ancestor of the same type. This makes an assumption that repairs are always maximally local, which probably does not hurt accuracy, since most repairs actually are quite short. However, this assumption is obviously not true in the general case, since in Figure 3 for example, the repair could trace all the way back to the S label at the root of the tree in the case of a restarted sentence. It is even possible that this implicit incentive to short repairs is responsible for some of the accuracy gains by discounting long repairs. In any case, future work will attempt to maintain the motivation behind the REPAIRED-INT modification while relaxing hard assumptions about repair distance.

## 4 Conclusion

This paper introduced three potential syntactic representations for speech with repairs, based on the idea that errors are not recognized as such until a correction is begun. The main result is a new representation, REPAIRED-INT, which, when transformed via the right-corner transform, makes a very attractive model for speech with repairs. This representation leads to a parser that improves on other parsing approaches in both overall parsing accuracy and accuracy recognizing words that have been edited.

## References

John J. Godfrey, Edward C. Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Proc. ICASSP*, pages 517–520.

John Hale, Izhak Shafran, Lisa Yung, Bonnie Dorr, Mary Harper, Anna Krasnyanskaya, Matthew Lease, Yang Liu, Brian Roark, Matthew Snover, and Robin Stewart. 2006. PCFGs with syntactic and prosodic indicators of speech repairs. In *Proceedings of the 45th Annual Conference of the Association for Computational Linguistics (COLING-ACL)*.

Mark Johnson and Eugene Charniak. 2004. A tag-based noisy channel model of speech repairs. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL '04)*, pages 33–39, Barcelona, Spain.

Mark Johnson. 1998. PCFG models of linguistic tree representation. *Computational Linguistics*, 24:613–632.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430.

Willem J.M. Levelt. 1983. Monitoring and self-repair in speech. *Cognition*, 14:41–104.

Tim Miller and William Schuler. 2008. A unified syntactic model for parsing fluent and disfluent speech. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL '08)*.

Brian Roark and Mark Johnson. 1999. Efficient probabilistic top-down and left-corner parsing. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL 99)*.

Elizabeth Shriberg. 1994. *Preliminaries to a Theory of Speech Disfluencies*. Ph.D. thesis, University of California at Berkeley.