Reconstructing false start errors in spontaneous speech text

Erin Fitzgerald
Johns Hopkins University
Baltimore, MD, USA
erinf@jhu.edu

Keith Hall
Google, Inc.
Zurich, Switzerland
kbhall@google.com

Frederick Jelinek
Johns Hopkins University
Baltimore, MD, USA
jelinek@jhu.edu

Abstract

This paper presents a conditional random field-based approach for identifying speaker-produced disfluencies (i.e. if and where they occur) in spontaneous speech transcripts. We emphasize false start regions, which are often missed in current disfluency identification approaches as they lack lexical or structural similarity to the speech immediately following. We find that combining lexical, syntactic, and language model-related features with the output of a state-of-the-art disfluency identification system improves overall word-level identification of these and other errors. Improvements are reinforced under a stricter evaluation metric requiring exact matches between cleaned sentences annotator-produced reconstructions, and altogether show promise for general reconstruction efforts.

1 Introduction

The output of an automatic speech recognition (ASR) system is often not what is required for subsequent processing, in part because speakers themselves often make mistakes (e.g. stuttering, self-correcting, or using filler words). A cleaner speech transcript would allow for more accurate language processing as needed for natural language processing tasks such as machine translation and conversation summarization which often assume a grammatical sentence as input.

A system would accomplish reconstruction of its spontaneous speech input if its output were to represent, in flawless, fluent, and content-preserving text, the message that the speaker intended to convey. Such a system could also be applied not only to spontaneous English speech, but to correct common mistakes made by non-native speakers (Lee and Seneff, 2006), and possibly extended to non-English speaker errors.

A key motivation for this work is the hope that a cleaner, reconstructed speech transcript will allow for simpler and more accurate human and natural language processing, as needed for applications like machine translation, question answering, text summarization, and paraphrasing which often assume a grammatical sentence as input. This benefit has been directly demonstrated for statistical machine translation (SMT). Rao et al. (2007) gave evidence that simple disfluency removal from transcripts can improve BLEU (a standard SMT evaluation metric) up to 8% for sentences with disfluencies. The presence of disfluencies were found to hurt SMT in two ways: making utterances longer without adding semantic content (and sometimes adding false content) and exacerbating the data mismatch between the spontaneous input and the clean text training data.

While full speech reconstruction would likely require a range of string transformations and potentially deep syntactic and semantic analysis of the errorful text (Fitzgerald, 2009), in this work we will first attempt to resolve less complex errors, corrected by deletion alone, in a given manually-transcribed utterance.

We build on efforts from (Johnson et al., 2004), aiming to improve overall recall – especially of false start or non-copy errors – while concurrently maintaining or improving precision.

1.1 Error classes in spontaneous speech

Common simple disfluencies in sentence-like utterances (SUs) include *filler words* (i.e. "um", "ah", and discourse markers like "you know"), as well as speaker edits consisting of a *reparandum*, an *interruption point (IP)*, an optional *interregnum* (like "I mean"), and a *repair* region (Shriberg, 1994), as seen in Figure 1.

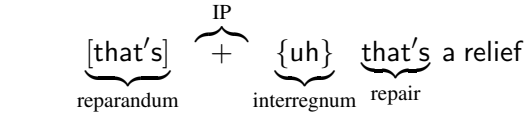

Figure 1: Typical edit region structure. In these and other examples, reparandum regions are in brackets ('[', ']'), interregna are in braces ('{', '}'), and interruption points are marked by '+'.

These reparanda, or *edit regions*, can be classified into three main groups:

1. In a **repetition** (above), the repair phrase is approximately identical to the reparandum.

2. In a **revision**, the repair phrase alters reparandum words to correct the previously stated thought.

   EX1: but [when he] + {i mean} when she put it that way

   EX2: it helps people [that are going to quit] + that would be quitting anyway

3. In a **restart fragment** (also called a false start), an utterance is aborted and then restarted with a new train of thought.

   EX3: and [i think he's] + he tells me he's glad he has one of those

   EX4: [amazon was incorporated by] {uh} well i only knew two people there

In *simple cleanup* (a precursor to full speech reconstruction), all detected filler words are deleted, and the reparanda and interregna are deleted while the repair region is left intact. This is a strong initial step for speech reconstruction, though more complex and less deterministic changes are often required for generating fluent and grammatical speech text.

In some cases, such as the repetitions mentioned above, simple cleanup is adequate for reconstruction. However, simply deleting the identified reparandum regions is not always optimal. We would like to consider preserving these fragments (for false starts in particular) if

1. the fragment contains content words, and

2. its information content is distinct from that in surrounding utterances.

In the first restart fragment example (EX3 in Section 1.1), the reparandum introduces no new active verbs or new content, and thus can be safely deleted. The second example (EX4) however demonstrates a case when the reparandum may be considered to have unique and preservable content of its own. Future work should address how to most appropriately reconstruct speech in this and similar cases; this initial work will for risk information loss as we identify and delete these reparandum regions.

1.2 Related Work

Stochastic approaches for simple disfluency detection use features such as lexical form, acoustic cues, and rule-based knowledge. Most state-of-the-art methods for edit region detection such as (Johnson and Charniak, 2004; Zhang and Weng, 2005; Liu et al., 2004; Honal and Schultz, 2005) model speech disfluencies as a noisy channel model. In a noisy channel model we assume that an unknown but fluent string F has passed through a disfluency-adding channel to produce the observed disfluent string D, and we then aim to recover the most likely input string $\hat{F}$, defined as

$$\begin{aligned} \hat{F} &= \text{argmax}_F P(F|D) \\ &= \text{argmax}_F P(D|F)P(F) \end{aligned}$$

where $P(F)$ represents a language model defining a probability distribution over fluent "source" strings F, and $P(D|F)$ is the channel model defining a conditional probability distribution of observed sentences D which may contain the types of construction errors described in the previous subsection. The final output is a word-level tagging of the error condition of each word in the sequence, as seen in line 2 of Figure 2.

The Johnson and Charniak (2004) approach, referred to in this document as JC04, combines the noisy channel paradigm with a tree-adjoining grammar (TAG) to capture approximately repeated elements. The TAG approach models the crossed word dependencies observed when the reparandum incorporates the same or very similar words in roughly the same word order, which JC04 refer to as a *rough copy*. Our version of this system does not use external features such as prosodic classes, as they use in Johnson et al. (2004), but otherwise appears to produce comparable results to those reported.

While much progress has been made in simple disfluency detection in the last decade, even top-performing systems continue to be ineffective at identifying words in reparanda. To better understand these problems and identify areas

Label	% of words	Precision	Recall	F-score
Fillers	5.6%	64%	59%	61%
Edit (reparandum)	7.8%	85%	68%	75%

Table 1: Disfluency detection performance on the SSR test subcorpus using JC04 system.

Label	% of edits	Recall
Rough copy (RC) edits	58.8%	84.8%
Non-copy (NC) edits	41.2%	43.2%
Total edits	100.0%	67.6%

Table 2: Deeper analysis of edit detection performance on the SSR test subcorpus using JC04 system.

```
1 he  that  's  uh that 's a relief
2  E    E    E  FL   -   - -    -
3 NC   RC   RC  FL   -   - -    -
```

Figure 2: Example of word class and refined word class labels, where - denotes a non-error, FL denotes a filler, E generally denotes reparanda, and RC and NC indicate rough copy and non-copy speaker errors, respectively.

for improvement, we used the top-performing[1] JC04 noisy channel TAG edit detector to produce edit detection analyses on the test segment of the Spontaneous Speech Reconstruction (SSR) corpus (Fitzgerald and Jelinek, 2008). Table 1 demonstrates the performance of this system for detecting filled pause fillers, discourse marker fillers, and edit words. The results of a more granular analysis compared to a hand-refined reference (as shown in line 3 of Figure 2) are shown in Table 2. The reader will recall that precision P is defined as $P = \frac{|\text{correct}|}{|\text{correct}|+|\text{false}|}$ and recall $R = \frac{|\text{correct}|}{|\text{correct}|+|\text{miss}|}$. We denote the harmonic mean of P and R as F-score F and calculate it $F = \frac{2}{1/P+1/R}$.

As expected given the assumptions of the TAG approach, JC04 identifies repetitions and most revisions in the SSR data, but less successfully labels false starts and other speaker self-interruptions which do not have a cross-serial correlations. These non-copy errors (with a recall of only 43.2%), are hurting the overall edit detection recall score. Precision (and thus F-score) cannot be calculated for the experiment in Table 2; since the JC04 does not explicitly label edits as rough copies or non-copies, we have no way of knowing whether words falsely labeled as edits would have been considered as false RCs or false NCs. This will unfortunately hinder us from using JC04 as a direct baseline comparison in our work targeting false starts; however, we consider these results to be further motivation for the work.

Surveying these results, we conclude that there is still much room for improvement in the field of simple disfluency identification, especially the cases of detecting non-copy reparandum and learning how and where to implement non-deletion reconstruction changes.

2 Approach

2.1 Data

We conducted our experiments on the recently released Spontaneous Speech Reconstruction (SSR) corpus (Fitzgerald and Jelinek, 2008), a medium-sized set of disfluency annotations atop Fisher conversational telephone speech (CTS) data (Cieri et al., 2004). Advantages of the SSR data include

- aligned parallel original and cleaned sentences
- several levels of error annotations, allowing for a coarse-to-fine reconstruction approach
- multiple annotations per sentence reflecting the occasional ambiguity of corrections

As reconstructions are sometimes non-deterministic (illustrated in EX6 in Section 1.1), the SSR provides two manual reconstructions for each utterance in the data. We use these dual annotations to learn complementary approaches in training and to allow for more accurate evaluation.

The SSR corpus does not explicitly label all reparandum-like regions, as defined in Section 1.1, but only those which annotators selected to delete.

[1] As determined in the RT04 EARS Metadata Extraction Task

Thus, for these experiments we must implicitly attempt to replicate annotator decisions regarding whether or not to delete reparandum regions when labeling them as such. Fortunately, we expect this to have a negligible effect here as we will emphasize utterances which do not require more complex reconstructions in this work.

The Spontaneous Speech Reconstruction corpus is partitioned into three subcorpora: 17,162 training sentences (119,693 words), 2,191 sentences (14,861 words) in the development set, and 2,288 sentences (15,382 words) in the test set. Approximately 17% of the total utterances contain a reparandum-type error.

The output of the JC04 model ((Johnson and Charniak, 2004) is included as a feature and used as an approximate baseline in the following experiments. The training of the TAG model within this system requires a very specific data format, so this system is trained not with SSR but with Switchboard (SWBD) (Godfrey et al., 1992) data as described in (Johnson and Charniak, 2004). Key differences in these corpora, besides the form of their annotations, include:

- SSR aims to correct speech output, while SWBD edit annotation aims to identify reparandum structures specifically. Thus, as mentioned, SSR only marks those reparanda which annotators believe must be deleted to generate a grammatical and content-preserving reconstruction.
- SSR considers some phenomena such as leading conjunctions ("and i did" → "i did") to be fillers, while SWBD does not.
- SSR includes more complex error identification and correction, though these effects should be negligible in the experimental setup presented herein.

While we hope to adapt the trained JC04 model to SSR data in the future, for now these difference in task, evaluation, and training data will prevent direct comparison between JC04 and our results.

2.2 Conditional random fields

Conditional random fields (Lafferty et al., 2001), or CRFs, are undirected graphical models whose prediction of a hidden variable sequence Y is globally conditioned on a given observation sequence X, as shown in Figure 3. Each observed

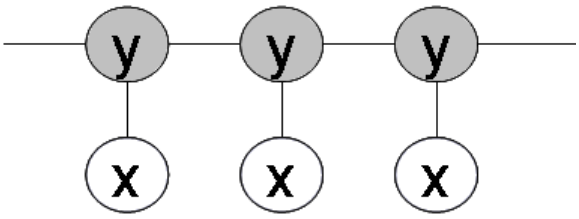

Figure 3: Illustration of a conditional random field. For this work, **x** represents observable inputs for each word as described in Section 3.1 and **y** represents the error class of each word (Section 3.2).

state $x_i \in X$ is composed of the corresponding word w_i and a set of additional features F_i, detailed in Section 3.1.

The conditional probability of this model can be represented as

$$p_\Lambda(Y|X) = \frac{1}{Z_\lambda(X)} \exp(\sum_k \lambda_k F_k(X,Y)) \quad (1)$$

where $Z_\lambda(X)$ is a global normalization factor and $\Lambda = (\lambda_1 \ldots \lambda_K)$ are model parameters related to each feature function $F_k(X,Y)$.

CRFs have been widely applied to tasks in natural language processing, especially those involving tagging words with labels such as part-of-speech tagging and shallow parsing (Sha and Pereira, 2003), as well as sentence boundary detection (Liu et al., 2005; Liu et al., 2004). These models have the advantage that they model sequential context (like hidden Markov models (HMMs)) but are discriminative rather than generative and have a less restricted feature set. Additionally, as compared to HMMs, CRFs offer conditional (versus joint) likelihood, and directly maximizes posterior label probabilities $P(E|O)$.

We used the GRMM package (Sutton, 2006) to implement our CRF models, each using a zero-mean Gaussian prior to reduce over-fitting our model. No feature reduction is employed, except where indicated.

3 Word-Level ID Experiments

3.1 Feature functions

We aim to train our CRF model with sets of features with orthogonal analyses of the errorful text, integrating knowledge from multiple sources. While we anticipate that repetitions and other rough copies will be identified primarily by lexical

and local context features, this will not necessarily help for false starts with little or no lexical overlap between reparandum and repair. To catch these errors, we add both language model features (trained with the SRILM toolkit (Stolcke, 2002) on SWBD data with EDITED reparandum nodes removed), and syntactic features to our model. We also included the output of the JC04 system – which had generally high precision on the SSR data – in the hopes of building on these results.

Altogether, the following features F were extracted for each observation x_i.

- **Lexical features**, including
  - the lexical item and part-of-speech (POS) for tokens t_i and t_{i+1},
  - distance from previous token to the next matching word/POS,
  - whether previous token is partial word and the distance to the next word with same start, and
  - the token's (normalized) position within the sentence.
- **JC04-edit**: whether previous, next, or current word is identified by the JC04 system as an edit and/or a filler (fillers are classified as described in (Johnson et al., 2004)).
- **Language model features**: the unigram log probability of the next word (or POS) token $p(t)$, the token log probability conditioned on its multi-token history h $(p(t|h))$[2], and the log ratio of the two $(log\frac{p(t|h)}{p(t)})$ to serve as an approximation for mutual information between the token and its history, as defined below.

$$I(t;h) = \sum_{h,t} p(h,t) \log \frac{p(h,t)}{p(h)p(t)}$$
$$= \sum_{h,t} p(h,t) \left[\log \frac{p(t|h)}{p(t)}\right]$$

  This aims to capture unexpected n-grams produced by the juxtaposition of the reparandum and the repair. The mutual information feature aims to identify when common words are seen in uncommon context (or, alternatively, penalize rare n-grams normalized for rare words).

[2]In our model, word historys h encompassed the previous two words (a 3-gram model) and POS history encompassed the previous four POS labels (a 5-gram model)

- **Non-terminal (NT) ancestors**: Given an automatically produced parse of the utterance (using the Charniak (1999) parser trained on Switchboard (SWBD) (Godfrey et al., 1992) CTS data), we determined for each word all NT phrases just completed (if any), all NT phrases about to start to its right (if any), and all NT constituents for which the word is included.

  (Ferreira and Bailey, 2004) and others have found that false starts and repeats tend to end at certain points of phrases, which we also found to be generally true for the annotated data.

Note that the syntactic and POS features we used are extracted from the output of an automatic parser. While we do not expect the parser to always be accurate, especially when parsing errorful text, we hope that the parser will at least be consistent in the types of structures it assigns to particular error phenomena. We use these features in the hope of taking advantage of that consistency.

3.2 Experimental setup

In these experiments, we attempt to label the following word-boundary classes as annotated in SSR corpus:

- fillers (FL), including filled pauses and discourse markers (∼5.6% of words)
- rough copy (RC) edit (reparandum incorporates the same or very similar words in roughly the same word order, including repetitions and some revisions) (∼4.6% of words)
- non-copy (NC) edit (a speaker error where the reparandum has no lexical or structural relationship to the repair region following, as seen in restart fragments and some revisions) (∼3.2% of words)

Other labels annotated in the SSR corpus (such as insertions and word reorderings), have been ignored for these error tagging experiments.

We approach our training of CRFs in several ways, detailed in Table 3. In half of our experiments (#1, 3, and 4), we trained a single model to predict all three annotated classes (as defined at the beginning of Section 3.3), and in the other half (#2, 5, and 6), we trained the model to predict NCs only, NCs and FLs, RCs only, or RCs and FLs (as FLs often serve as interregnum, we predict that these will be a valuable cue for other edits).

Setup	Train data	Test data	Classes trained per model
#1	Full train	Full test	FL + RC + NC
#2	Full train	Full test	{RC,NC}, FL+{RC,NC}
#3	Errorful SUs	Errorful SUs	FL + RC + NC
#4	Errorful SUs	Full test	FL + RC + NC
#5	Errorful SUs	Errorful SUs	{RC,NC}, FL+{RC,NC}
#6	Errorful SUs	Full test	{RC,NC}, FL+{RC,NC}

Table 3: Overview of experimental setups for word-level error predictions.

We varied the subcorpus utterances used in training. In some experiments (#1 and 2) we trained with the entire training set[3], including sentences without speaker errors, and in others (#3-6) we trained only on those sentences containing the relevant deletion errors (and no additionally complex errors) to produce a densely errorful training set. Likewise, in some experiments we produced output only for those test sentences which we knew to contain simple errors (#3 and 5). This was meant to emulate the ideal condition where we could perfectly predict which sentences contain errors before identifying where exactly those errors occurred.

The JC04-edit feature was included to help us build on previous efforts for error classification. To confirm that the model is not simply replicating these results and is indeed learning on its own with the other features detailed, we also trained models without this JC04-edit feature.

3.3 Evaluation of word-level experiments

3.3.1 Word class evaluation

We first evaluate edit detection accuracy on a per-word basis. To evaluate our progress identifying word-level error classes, we calculate precision, recall and F-scores for each labeled class c in each experimental scenario. As usual, these metrics are calculated as ratios of correct, false, and missed predictions. However, to take advantage of the double reconstruction annotations provided in SSR (and more importantly, in recognition of the occasional ambiguities of reconstruction) we modified these calculations slightly as shown below.

$$\text{corr}(c) = \sum_{i:c_{w_i}=c} \delta(c_{w_i} = c_{g_1,i} \text{ or } c_{w_i} = c_{g_2,i})$$

$$\text{false}(c) = \sum_{i:c_{w_i}=c} \delta(c_{w_i} \neq c_{g_1,i} \text{ and } c_{w_i} \neq c_{g_2,i})$$

$$\text{miss}(c) = \sum_{i:c_{g_1,i}=c} \delta(c_{w_i} \neq c_{g_1,i})$$

where c_{w_i} is the hypothesized class for w_i and $c_{g_1,i}$ and $c_{g_2,i}$ are the two reference classes.

Setup	Class labeled	FL	RC	NC
Train and test on all SUs in the subcorpus				
#1	FL+RC+NC	**71.0**	80.3	47.4
#2	NC	-	-	42.5
#2	NC+FL	70.8	-	**47.5**
#2	RC	-	84.2	-
#2	RC+FL	67.8	**84.7**	-
Train and test on errorful SUs				
#3	FL+RC+NC	91.6	84.1	52.2
#4	FL+RC+NC	44.1	69.3	31.6
#5	NC	-	-	73.8
#6	*w/ full test*	-	-	39.2
#5	NC+FL	90.7	-	69.8
#6	*w/ full test*	50.1	-	38.5
#5	RC	-	88.7	-
#6	*w/ full test*	-	75.0	-
#5	RC+FL	92.3	87.4	-
#6	*w/ full test*	62.3	73.9	-

Table 4: Word-level error prediction F_1-score results: Data variation. The first column identifies which data setup was used for each experiment (Table 3). The highest performing result for each class in the first set of experiments has been highlighted.

Analysis: Experimental results can be seen in Tables 4 and 5. Table 4 shows the impact of

[3]Using both annotated SSR reference reconstructions for each utterance

Features	FL	RC	NC
JC04 only	56.6	69.9-81.9	1.6-21.0
lexical only	56.5	72.7	33.4
LM only	0.0	15.0	0.0
NT bounds only	44.1	35.9	11.5
All but JC04	58.5	79.3	33.1
All but lexical	66.9	76.0	19.6
All but LM	67.9	83.1	41.0
All but NT bounds	61.8	79.4	33.6
All	**71.0**	**80.3**	**47.4**

Table 5: Word-level error prediction F-score results: Feature variation. All models were trained with experimental setup #1 and with the set of features identified.

training models for individual features and of constraining training data to contain only those utterances known to contain errors. It also demonstrates the potential impact on error classification after prefiltering test data to those SUs with errors. Table 5 demonstrates the contribution of each group of features to our CRF models.

Our results demonstrate the impact of varying our training data and the number of label classes trained for. We see in Table 4 from setup #5 experiments that training and testing on error-containing utterances led to a dramatic improvement in F_1-score. On the other hand, our results for experiments using setup #6 (where training data was filtered to contain errorful data but test data was fully preserved) are consistently worse than those of either setup #2 (where both train and test data was untouched) or setup #5 (where both train and test data were prefiltered). The output appears to suffer from sample bias, as the prior of an error occurring in training is much higher than in testing. This demonstrates that a densely errorful training set alone cannot improve our results when testing data conditions do not match training data conditions. However, efforts to identify errorful sentences before determining where errors occur in those sentences may be worthwhile in preventing false positives in error-less utterances.

We next consider the impact of the four feature groups on our prediction results. The CRF model appears competitive even without the advantage of building on JC04 results, as seen in Table 5[4]. Interestingly and encouragingly, the NT bounds features which indicate the linguistic phrase structures beginning and ending at each word according to an automatic parse were also found to be highly contribututive for both fillers and non-copy identification. We believe that further pursuit of syntactic features, especially those which can take advantage of the context-free weakness of statistical parsers like (Charniak, 1999) will be promising in future research.

It was unexpected that NC classification would be so sensitive to the loss of lexical features while RC labeling was generally resilient to the dropping of any feature group. We hypothesize that for rough copies, the information lost from the removal of the lexical items might have been compensated for by the JC04 features as JC04 performed most strongly on this error type. This should be further investigated in the future.

3.3.2 Strict evaluation: SU matching

Depending on the downstream task of speech reconstruction, it could be imperative not only to identify many of the errors in a given spoken utterance, but indeed to identify *all* errors (and only those errors), yielding the precise cleaned sentence that a human annotator might provide.

In these experiments we apply *simple cleanup* (as described in Section 1.1) to both JC04 output and the predicted output for each experimental setup in Table 3, deleting words when their right boundary class is a filled pause, rough copy or non-copy.

Taking advantage of the dual annotations for each sentence in the SSR corpus, we can report both single-reference and double-reference evaluation. Thus, we judge that if a hypothesized cleaned sentence exactly matches *either* reference sentence cleaned in the same manner, we count the cleaned utterance as correct and otherwise assign no credit.

Analysis: We see the outcome of this set of experiments in Table 6. While the unfiltered test sets of JC04-1, setup #1 and setup #2 appear to have much higher sentence-level cleanup accuracy than the other experiments, we recall that this is natural also due to the fact that the majority of these sentences should not be cleaned at all, besides

[4]JC04 results are shown as a range for the reasons given in Section 1.2: since JC04 does not on its own predict whether an "edit" is a rough copy or non-copy, it is impossible to calculate precision and thus F_1 score precisely. Instead, here we show the resultant F_1 for the best case and worst case precision range.

Setup	Classes deleted	# SUs	# SUs which match gold	% accuracy
Baseline	only filled pauses	2288	1800	78.7%
JC04-1	E+FL	2288	1858	81.2%
CRF-#1	RC, NC, and FL	2288	1922	84.0%
CRF-#2	$\bigcup$ {RC,NC}	2288	1901	83.1%
Baseline	only filled pauses	281	5	1.8%
JC04-2	E+FL	281	126	44.8%
CRF-#3	RC, NC, and FL	281	156	55.5%
CRF-#5	$\bigcup$ {RC,NC}	281	132	47.0%

Table 6: Word-level error predictions: exact SU match results. JC04-2 was run only on test sentences known to contain some error to match the conditions of Setup #3 and #5 (from Table 3). For the baselines, we delete only filled pause filler words like "eh" and "um".

occasional minor filled pause deletions. Looking specifically on cleanup results for sentences known to contain at least one error, we see, once again, that our system outperforms our baseline JC04 system at this task.

4 Discussion

Our first goal in this work was to focus on an area of disfluency detection currently weak in other state-of-the-art speaker error detection systems – false starts – while producing comparable classification on repetition and revision speaker errors. Secondly, we attempted to quantify how far deleting identified edits (both RC and NC) and filled pauses could bring us to full reconstruction of these sentences.

We've shown in Section 3 that by training and testing on data prefiltered to include only utterances with errors, we can dramatically improve our results, not only by improving identification of errors but presumably by reducing the risk of falsely predicting errors. We would like to further investigate to understand how well we can automatically identify errorful spoken utterances in a corpus.

5 Future Work

This work has shown both achievable and demonstrably feasible improvements in the area of identifying and cleaning simple speaker errors. We believe that improved sentence-level identification of errorful utterances will help to improve our word-level error identification and overall reconstruction accuracy; we will continue to research these areas in the future. We intend to build on these efforts, adding prosodic and other features to our CRF and maximum entropy models,

In addition, as we improve the word-level classification of rough copies and non-copies, we will begin to move forward to better identify more complex speaker errors such as missing arguments, misordered or redundant phrases. We will also work to apply these results directly to the output of a speech recognition system instead of to transcripts alone.

Acknowledgments

The authors thank our anonymous reviewers for their valuable comments. Support for this work was provided by NSF PIRE Grant No. OISE-0530118. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the supporting agency.

References

J. Kathryn Bock. 1982. Toward a cognitive psychology of syntax: Information processing contributions to sentence formulation. *Psychological Review*, 89(1):1–47, January.

Eugene Charniak. 1999. A maximum-entropy-inspired parser. In *Meeting of the North American Association for Computational Linguistics*.

Christopher Cieri, Stephanie Strassel, Mohamed Maamouri, Shudong Huang, James Fiumara, David Graff, Kevin Walker, and Mark Liberman. 2004. Linguistic resource creation and distribution for EARS. In *Rich Transcription Fall Workshop*.

Fernanda Ferreira and Karl G. D. Bailey. 2004. Disfluencies and human language comprehension. *Trends in Cognitive Science*, 8(5):231–237, May.

Erin Fitzgerald and Frederick Jelinek. 2008. Linguistic resources for reconstructing spontaneous speech text. In *Proceedings of the Language Resources and Evaluation Conference*, May.

Erin Fitzgerald. 2009. *Reconstructing Spontaneous Speech*. Ph.D. thesis, The Johns Hopkins University.

John J. Godfrey, Edward C. Holliman, and Jane McDaniel. 1992. SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 517–520, San Francisco.

Matthias Honal and Tanja Schultz. 2005. Automatic disfluency removal on recognized spontaneous speech – rapid adaptation to speaker-dependent disfluenices. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*.

Mark Johnson and Eugene Charniak. 2004. A TAG-based noisy channel model of speech repairs. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Mark Johnson, Eugene Charniak, and Matthew Lease. 2004. An improved model for recognizing disfluencies in conversational speech. In *Rich Transcription Fall Workshop*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA.

John Lee and Stephanie Seneff. 2006. Automatic grammar correction for second-language learners. In *Proceedings of the International Conference on Spoken Language Processing*.

Yang Liu, Elizabeth Shriberg, Andreas Stolcke, Barbara Peskin, and Mary Harper. 2004. The ICSI/UW RT04 structural metadata extraction system. In *Rich Transcription Fall Workshop*.

Yang Liu, Andreas Stolcke, Elizabeth Shriberg, and Mary Harper. 2005. Using conditional random fields for sentence boundary detection in speech. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 451–458, Ann Arbor, MI.

Sharath Rao, Ian Lane, and Tanja Schultz. 2007. Improving spoken language translation by automatic disfluency removal: Evidence from conversational speech transcripts. In *Machine Translation Summit XI*, Copenhagen, Denmark, October.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *HLT-NAACL*.

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

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Denver, CO, September.

Charles Sutton. 2006. GRMM: A graphical models toolkit. http://mallet.cs.umass.edu.

Qi Zhang and Fuliang Weng. 2005. Exploring features for identifying edited regions in disfluent sentences. In *Proceedings of the International Workshop on Parsing Techniques*, pages 179–185.