A Sequential Repetition Model for Improved Disfluency Detection

Mari Ostendorf[1], *Sangyun Hahn*[2]

[1]Electrical Engineering Department, [2]Computer Science & Engineering Department
University of Washington, Seattle, WA, USA
mo@ee.washington.edu, hahnsy@gamil.com

Abstract

This paper proposes a new method for automatically detecting disfluencies in spontaneous speech – specifically, self-corrections – that explicitly models repetitions vs. other disfluencies. We show that, in a corpus of Supreme Court oral arguments, repetition disfluencies can be longer and more stutter-like than the short repetitions observed in the Switchboard corpus and suggest that they can be better represented with a flat structure that covers the full sequence. Since these disfluencies are relatively easy to detect, weakly supervised training is an effective way to minimize labeling costs. By explicitly modeling these, we improve general disfluency detection within and across domains, and we provide a richer transcript.

Index Terms: disfluencies, stuttering, rich transcription

1. Introduction

Speech disfluencies – including filled pauses, repetitions, repairs and false starts – are frequent events in spontaneous speech. While the rate of disfluencies varies with the speaker and context, one study observed disfluencies once in every 20 words, affecting up to one third of utterances [1]. Disfluencies are important to account for, both because of the information that they provide about the speaker and because of the challenge that the disrupted grammatical flow poses for natural language processing of spoken transcripts. In parsing, for example, explicit modeling of disfluencies in decoding improves parser accuracy even when the parser is trained on spontaneous speech transcripts (vs. text) [2, 3], and clean-up of disfluencies before training the parser in combination with disfluency detection gives further gains [4]. In this paper, we use categorization of disfluency types to improve automatic detection and to provide a richer representation of the transcript.

Using the terminology introduced in [1], speech disfluencies have the following basic pattern:

[reparandum + { interregnum } repair]

where the reparandum is the portion of the utterance that is corrected or abandoned, + indicates the interruption point (IP) which marks the end of reparandum, the (optional) interregnum could include filled pauses, edit phrases (e.g. “I mean”), or other filler words (e.g. “you know”), and the repair (also optional) is the portion of the utterance that corrects the reparandum. When the repair is omitted, the disfluency is referred to as a deletion, restart, or false start.

Some examples of disfluencies from our corpora include:

[I just + I] enjoy working ...
[we + you’d] have to just ...
[it’s + { uh } it’s] almost like ...
[by +] it was attached to ...
[we want + { well } in our area we want] to ...
[and I’m wondering [if + if] + and] your argument ...
[the + [th- + the]] DEA ...

where the last two cases are referred to as complex disfluencies because they include multiple or nested disfluencies. The last example includes a word fragment, frequently observed in disfluencies but often missed in both human and automatic transcription.

In order to automate the detection of speech disfluencies, various approaches have been studied, primarily focusing on self-corrections since filled pauses are trivial to detect. While prosodic features are useful for detecting interruption points [5, 6, 7, 8], methods using only lexical features do almost as well as those with both prosodic and lexical features, and many studies have leveraged only lexical features. In addition, prosodic features require reasonably good word time alignments, which are not available for one corpus used in our study, so our focus is on detection from features derived from the word transcript.

There are a few different approaches for detecting disfluencies using only word transcripts. N-gram language models have been used in a variety of ways to detect disfluencies [9, 10, 11, 12]. Other studies detect disfluencies jointly with finding parse structure [2, 3, 13, 14, 15, 16, 17]. The idea of using a noisy channel model to represent the differences between reparandum and repair is explored in both types of approaches [11, 12, 3, 17]. Discriminative approaches were first explored in [8], and the method using conditional random fields (CRFs) was extended to include constraints via integer linear programming (ILP) in [18]. It is difficult to compare the various approaches because of differences in text processing, data and evaluation methodology, but CRFs give some of the best reported results, so we use that approach in this work.

Most disfluency detection work aims at detecting the reparandum without explicitly distinguishing between different types. However, different types of self-correction disfluencies occur at different frequencies (repetitions are most frequent), and they reflect speaker variability (some speakers are “repeaters” and others are “deleters”) [1]. Studies have also suggested that different types of disfluencies play different roles. For example, filled pauses are often associated with planning and floor holding [19]. It has been suggested that repetition disfluencies are “covert” prearticulatory repairs [20]. Production studies using pictures with high vs. low name agreement and controlling for word frequencies found differences in the types of disfluencies that are observed in contexts with low name agreement vs. low frequency article gender [21].

Motivated by these studies and our own corpus analyses, this work explores explicit modeling of repetitions separately from repairs and false starts. We propose an alternative (flat) analysis of sequential repetitions and provide examples and statistics from two corpora to support the approach (Section 2). Since hand-labeled data for the different types with the flat analysis are not available in a standard labeled corpus, we use weakly supervised learning to train the detection model given a small amount of type-labeled data in combination with a larger training set with the standard analysis (Section 3). Improved disfluency detection is obtained for both in-domain and cross-domain detection experiments (Section 4). The paper concludes with a summary and suggestions for future work (Section 5).

2. Corpus Analyses

2.1. Corpora

This study looks at disfluencies in two speech sources: the Switchboard corpus of telephone conversations, and three cases from the archive of oral arguments of the Supreme Court of the United States (SCOTUS).

The Switchboard corpus [22] includes 5-10 minute conversations between strangers on pre-specified topics. It is the most frequently used data for disfluency detection studies because of the large amount of hand-annotated data available (over 1M words). The data was hand-annotated for disfluency structure by the Linguistics Data Consortium (LDC), using the bracketing structure in the examples above (reparandum, interruption point, repair) augmented with labels in the interregnum (e.g., filler, discourse marker, explicit edit term). A subset of 51 conversations were transformed to the flattened repetition structure described in the next section and repetition type labels were added. The training subset (40 conversations) was used in the distributional studies below. While it is difficult to compare results directly because of differences in text pre-processing and test set definitions, some of the best reported results are 83.8 F-score for detection of reparandum (edit) words [12], and F-scores of 80 and 83 for detecting reparandum onsets and interruption points, respectively [18].

The SCOTUS archive (available via oyez.org) includes over 50 years of Supreme Court cases. Each oral argument includes statements, questions and responses between justices and advocates, with recordings lasting roughly an hour. The transcripts available for the full collection have not been carefully annotated to preserve disfluencies, but hand-corrected transcripts are available for some cases. The transcripts of three cases were analyzed in this study (comprising 28.2k words), all hand corrected by University of Washington undergraduate students. The three cases were then hand-annotated for disfluencies using the modified annotation scheme described next. The data was only used for testing and for extracting the distributional statistics below. A key difference between the two data sets is that the SCOTUS data reflect a much higher stakes context.

2.2. Repetition Disfluency Annotation

Repetition disfluencies are generally defined to include exact repetitions of a word or a phrase, as in *[I + I]* or *[it's a + it's a].* In some cases they may involve word fragment, as in *[wh- + what].* Excluding filled pauses, repetition disfluencies represent the largest class of disfluencies in human-human conversations [1]. While most repetitions are simple disfluencies (i.e. the word/phrase is repeated only once), longer repetitions are not unusual. *"I I I"* is the most frequent trigram in Switchbard, and there are many instances of four-word repetitions in the SCOTUS corpus. These repetitions can be stutter-like, as in: *[than um a + th- + th- + than a].* Like [23], we think that the binary branching representation of these as complex disfluencies obscures what appears to be simply multiple attempts to produce a target phrase, as in *[non- + nonmed- + nonmedical].* However, our strategy for annotating these cases stays closer to the Switchboard disfluency annotation framework. Specifically, we remove nested bracketing within a disfluency as long as each repetition is a copy or a prefix substring of the final correction. The result is a flattened structure with multiple IPs and an "S" associated with the bracket to indicate a repetition or stutter-like disfluency. The advantage of the flatter structure is that the connection between each instance of the repetition is more clear when there are incomplete attempts, as in the first examples of Table 1. For mixed regions of repetitions and repairs, or for partial repetitions that match a later part of the phrase, the nested disfluency annotation is retained. In the original Switchboard annotation, ambiguities of complex disfluency structure are resolved by prefering a left branching structure, e.g. *[[he + she] + she]* vs. *[he + [S she + she]],* a preference to group repetitions in the annotation here. More examples of the mapping are given in Table 1.

Note that cases where the repair inserts a word before the repetition, as in *[de- + as described],* would not be labeled as a repetition disfluency.

2.3. Distributional Characteristics

The examples above are all taken from the SCOTUS data, where there were many more instances for which the revised annotation framework seemed useful. Looking at the subset of Switchboard and SCOTUS data annotated for these repetition regions, we find that the rate of false starts is similar in both corpora (5-6%), but the SCOTUS corpus has a much higher rate of repetition disfluencies than the Switchboard corpus (70% vs. 63%, respectively), counting a multiple repetition sequence as one disfluency. In addition, the SCOTUS corpus includes many more instances having greater than one repetition, so that the average number of interruption points in a repetition is greater than in Switchboard: 1.18 vs. 1.07, respectively. Anecdotally, we also noted several cases in the SCOTUS corpus where a phrase of 4-6 words was repeated, which was not observed in the Switchboard subset analyzed. Ignoring the type of disfluency, the reparandum is longer on average in the SCOTUS corpus than in Switchboard, with roughly 7% vs. 5% of spoken words associated with a reparandum (excluding filled pauses). These differences could reflect the higher stakes nature of the SCOTUS corpus, or it could be related to the fact that there are a smaller number of speakers in the SCOTUS data and inter-speaker variability is known to be substantial.

3. Automatic Disfluency Detection

3.1. Model Framework

Our goal in disfluency detection is to identify words in reparanda, i.e. words that would be deleted to make the transcript fluent. Our baseline disfluency detection model uses a CRF implemented with CRF++[1] with five class labels similar to tagging models used for named entity detection: begin edit (BE), inside edit (IE), last word in edit (IP), single-word reparandum (BE_IP), and a single other (O) label for words in fluent regions (including the repair), as in [18]. For complex disfluencies, we ignore the nesting structure but represent multiple interruption points and reparanda before the repair. Thus,

[1] CRF++ is available at http://code.google.com/p/crfpp/downloads/list.

[than um a + [[th- + th-] + than a]]	*[S than um a + th- + th- + than a]*
[a req- + [[a + a] + a] requirement]	*[S a req- + a + a + a requirement]*
[[[we'll + we'll] + we'll leave it] + we'll leave it]	*[S we'll + we'll + we'll leave it + we'll leave it]*
[[And + and] of course this, + and of course [[th- + this] + this]]	*[S And + and of course this, + and of course [S th- + this + this]]*

Table 1: Examples of reannotation of repetition disfluencies.

we can distinguish between sequential disfluencies, but not internal structure of a complex disfluency. We tried using an additional label for filled pauses, but it gave no performance gains.

As mentioned above, we focused on word-based features only, using hand transcriptions. We use a total of 140 feature types, including:

Word and word class features:

- n-gram (order 1-4) features over a window of ±3 words around the target word
- Part-of-speech tag features: tag n-grams similar to word n-grams.
- Whether the previous, current, or next word is a filled pause or discourse marker.
- Whether the current word is incomplete (fragment).

Pattern match features:

- Whether a word w_i and/or bigram $< w_i, w_{i+1} >$ is repeated anywhere within a given window (6 words).
- Whether a tag bigram $< t_i, t_{i+1} >$ is repeated anywhere within in a given window (6 words).
- Whether a current word or bigram repeats a preceding word or bigram anywhere within a given window (6 words).
- Whether the current word or bigram is repeated both in a preceding and following position within a given window (±6 words)

A frequency cutoff of 3 is used for all features, and l_2 regularization is used in training.

The indicator features for filled pauses and discourse markers are potentially useful for indicating the end of a reparandum, since they are often used in an interregnum. However, since they are also used frequently in other contexts, we did not find this feature to be very useful. As shown in the examples, word fragments are reliable cues to an interruption point when they are present. Since it is difficult to detect word fragments in speech recognition, many studies have omitted these cues. However, our goal is to automatically annotate hand transcripts, in which case the word fragments are useful, improving the detection of IPs by roughly 1.5 absolute F-score.

For explicit repetition modeling, we expand the label space to have separate states for stutter-like disfluencies and other disfluencies, e.g. BE_s and BE_o, leading to 9 classes. While expanding the state space could potentially lead to data sparsity issues, the expanded space can benefit from the fact that the pattern matching lexical cues to repetitions are quite different from repairs and false starts.

3.2. Learning Strategies

For training, we used the Switchboard corpus which contain hand-labeled disfluency markups. We held out 172 conversations for testing and 77 for development tuning; the remaining 877 conversations are used for training. Of the 51 reannotated conversations, 40 were used for training and 11 for testing. The tuning configuration identified for the standard (5-class) annotation was also used for the 9-class condition, so no augmented annotations were used for tuning. We trained models varying: i) the use of explicit types or not (9 vs. 5 classes), ii) the use of different amounts of training data (40 vs. 877 conversations), and iii) whether/how labels were used for the full training set. The different variations are listed in Table 2, with divisions indicating the amount of labeled training data available. In supervised training (sup), hand labeled data is available for all classes. Only a small subset has hand labels for the 9-class case; the data with the 5-class labels is referred to as the full set. In semi-supervised training (semi), hand labels are only used for the small set, and the remainder of the full set is treated as unlabeled. The particular approach used here is self-training, in which the unlabeled data is automatically labeled with a model trained on the hand-labeled small set. In weakly supervised training (weak), we use the 5-class hand-labeled data associated with the full set but augment it with automatically derived repetition labels that are constrained to respect constraints of the 5-class labels. Specifically, a CRF model (const) is trained to predict the 9-class labels using the small data set, given all the standard features above plus the 5-class labels as features.

Model (Data)	Labels	Learning
5-sup (small)	std	supervised
9-sup (small)	+ rep	supervised
9-const (small)	+ rep	supervised + constraints
5-sup (full)	std	supervised
9-weak (full)	+ rep	weakly supervised
5-semi (full)	+ rep, auto	semi-supervied
9-semi (full)	+ rep, auto	semi-supervied

Table 2: Models used for disfluency detection for standard (5 class) and repetition-augmented (9 class) annotations.

4. Experimental Results

4.1. Type-Independent Detection

The general disfluency detection results for the full test set (roughly 230k words) are given in Table 3 for the different model configurations. We report F-scores for detecting words in a reparandum, for comparison to the disfluency detection in parsing literature, and F-scores for detecting the onset of a reparandum (BE) and interruption points (IP), for comparison to the prior results reported with a CRF. With the caveat that the test sets and text processing are not directly comparable, our baseline results are roughly comparable to or better than previously reported results. The reparandum detection results are close: 82.8 vs. 83.8 from [12]. The BE and IP results are slightly better than the previously reported results for the CRF alone and the CRF combined with ILP [18]. A key reason why our CRF did much better than the previously reported results

	F-score		
Model (Data)	Reparandum	BE	IP
5-sup (small)	78.5	79.3	82.0
9-sup (small)	78.1	79.7	81.9
5-sup (full)	82.8	84.2	87.1
5-semi (full)	78.5	79.5	82.0
9-semi (full)	78.0	79.6	81.9
9-weak (full)	83.2	85.0	87.5

Table 3: Type-independent disfluency detection on Switchboard: with 5 vs. 9 disfluency classes and different training strategies.

	F-score		
Model (Data)	Repetition	Other	Either
9-sup (small)	85.0	32.4	73.8
9-const (small)	92.6	90.5	99.9
9-weak (full)	89.1	51.3	82.4

Table 4: Type-specific disfluency detection on Switchboard (repetition-annotated subset): with 5 vs. 9 disfluency classes and different training strategies.

for a CRF without ILP is due to the use of pattern matching features. Without these features, the BE and IP detection F-scores drop to 79.6 and 84.0, respectively.

For the two cases trained on the small training set, there are only small differences between the 5-class and the 9-class models, so the increase in the number of classes does not seem to be a problem for training. Using the full labeled training set improves detection of the reparandum by 3.7 F points for the 5-class model. The simple self-training approach to semi-supervised training does not give a significant gain over the supervised small training conditions for either the 5-class or the 9-class condition. For the weakly supervised training case, where we use the 5-class known labels as features to a CRF to constrain the 9-class labeling of the full training set, we achieve a small improvement in performance for the different tasks, with gains in both precision and recall.

4.2. Detection of Repetition Disfluencies

The performance of the three 9-class models on the hand-labeled test set (roughly 14k words) is summarized in Table 4, where we report F-measures for detecting repetitions, other disfluencies, and either category without distinction (words in the reparandum). As one would expect, the repetition disfluencies are much more reliably detected than other disfluencies, since the word match features almost always indicate a disfluency. We do observe some errors because some exact matches can be fluent, such as "that that," and subword matches with word fragments are not currently captured by our feature set. For the case where the hand-labeled disfluencies are used as features (thus eliminating false detections outside of the disfluency regions), the repetition and other disfluencies are detected very reliably (F-measures greater than 90), which is why weakly supervised training leads to performance gains. The detection of the reparandum is not 100% accurate, since the implementation uses the 5-class labels as features rather than as constraints. We find that weakly supervised training improves detection of both types of disfluencies, with the relative gain being larger for non-repetition disfluencies.

	F-score				
Model (Data)	Repetition	Other	Either	BE	IP
5-sup (small)	–	–	75.4	76.7	81.6
5-sup (full)	–	–	77.3	79.8	85.8
9-sup (small)	86.3	30.1	75.7	78.8	82.8
9-weak (full)	88.8	34.2	77.8	79.6	86.1

Table 5: Disfluency detection on SCOTUS: with 5 vs. 9 classes, and with different training strategies.

4.3. Corpus Comparisons

We also applied the models trained on the Switchboard corpus to the SCOTUS data (roughly 30k words), with results shown in Table 5. Comparing to Table 3, the BE detection performance degrades more than the IP detection for the 5-class model, though the difference is smaller when the training set is smaller. A similar finding is reported in a cross-domain study using a CRF with ILP [24]. Comparing to Table 4, we find that the performance of the small Switchboard-trained 9-class model actually has similar performance on the SCOTUS data as that observed for the Switchboard data, including that repetitions are much more reliably detected than other disfluencies. The 9-class modeled using weakly supervised training (i.e. more labeled Switchboard data) has improved performance for all categories compared to the small training set and improved reparandum and IP detection relative to the 5-class model trained on the full set.

5. Conclusions

In summary, this work proposes a modified annotation scheme for handling repetition (stutter-like) disfluencies, and shows that explicit modeling of such disfluencies leads to improved general disfluency detection. Because the repetitions are relatively easy to detect, weakly supervised learning methods are an effective way to reduce the cost of augmenting existing annotations. The weakly supervised training that uses the explicit repetition model also leads to improved performance in a cross-domain testing condition, benefiting both repetitions and repairs.

There are several possible directions for extending this work. First, the feature extraction could be improved to incorporate language model features from [12] and better leverage word fragments. Other detection frameworks may also be useful to explore for domain adaptation. In addition, it would be of interest to see if the explicit repetition model is useful in conjunction with parsing. Finally, because of prior work showing that speaker variability extends to relative frequency of different disfluency types, we anticipate that a computational model that distinguishes repetitions will facilitate speaker adaptation. The explicit type labeling may also be useful in studies of speaker variability in different human communication scenarios.

6. Acknowledgements

This work was supported in part by DARPA grant # FA8750-09-C-0179 and a Fulbright Fellowship. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Fulbright, DARPA or the U.S. Government.

7. References

[1] E. Shriberg, "Preliminaries to a theory of speech disfluencies," Ph.D. dissertation, Department of Psychology, University of California, Berkeley, CA, 1994.

[2] E. Charniak and M. Johnson, "Edit detection and parsing for transcribed speech," in *Proc. NAACL*, 2001, pp. 118–126.

[3] M. Johnson and E. Charniak, "A tag-based noisy channel model of speech repairs," in *Proc. ACL*, 2004.

[4] J. G. Kahn, M. Lease, E. Charniak, M. Johnson, and M. Ostendorf, "Effective use of prosody in parsing conversational speech," in *Proc. EMNLP-HLT*, 2005, pp. 233–240.

[5] C. Nakatani and J. Hirschberg, "A corpus-based study of repair cues in spontaneous speech," *Journal of the Acoustical Society of America*, pp. 1603–1616, 1994.

[6] E. Shriberg and A. Stolcke, "A prosody-only decision-tree model for disfluency detection," in *Eurospeech*, 1997, pp. 2383–2386.

[7] E. Shriberg, "Phonetic consequences of speech disfluency," in *Proc. International conference of Phonetics Sciences*, 1999, pp. 619–622.

[8] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper, "Enriching speech recognition with automatic detection of sentence boundaries and disfluencies," *IEEE Trans. Audio, Speech and Language Processing*, vol. 14, pp. 1526–1540, 2006.

[9] A. Stolcke and E. Shriberg, "Statistical language modeling for speech disfluencies," in *Proc. ICASSP*, 1996.

[10] P. Heeman and J. Allen, "Speech repairs, intonational phrases and discourse markers: Modeling speakersútterances in spoken dialogue," *Computational Linguistics*, vol. 25, pp. 527–571, 1999.

[11] M. Honal and T. Schultz, "Automatic disfluency removal on recognized spontaneous speech - rapid adaptation to speaker dependent disfluencies," in *Proc. ICASSP*, 2005.

[12] S. Zwarts and M. Johnson, "The impact of language models and los functions on repair disfluency detection," in *Proc. ACL*, 2011, pp. 703–711.

[13] M. Lease, M. Johnson, and E. Charniak, "Recognizing disfluencies in conversational speech," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 169–177, 2006.

[14] J. Hale, I. Shafran, L. Yung, B. Dorr, M. Harper, A. Krasnyanskaya, M. Lease, Y. Liu, B. Roark, M. Snover, and R. Stewart, "PCFGs with syntactic and prosodic indicators of speech repairs," in *Proc. Coling-ACL*, 2006.

[15] T. Miller, "Improved syntactic models for parsing speech with repairs," in *Proc. NAACL HLT*, 2009.

[16] T. Miller, L. Nguyen, and W. Schuler, "Parsing speech repair without specialized grammar symbols," in *Proc. ACL-IJCNLP*, 2009, pp. 277–280.

[17] S. Zwarts, M. Johnson, and R. Dale, "Detecting speech repairs incrementally using a noisy channel approach," in *Proc. Coling*, 2010, pp. 1371–1378.

[18] K. Georgila, "Using integer linear programming for detecting speech disfluencies," in *Proc. NAACL HLT*, 2009.

[19] E. Shriberg, "Spontaneous speech: How people really talk, and why engineers should care," in *Proc. Interspeech*, 2005.

[20] A. Postma and H. Kolk, "The covert repair hypothesis: Prearticulatory repair process in normal and stuttered disfluencies," *Journal os Speech and Hearing Research*, vol. 36, pp. 472–487, 1993.

[21] R. Hartsuiker and L. Notebaert, "Lexical access problems lead to disfluencies in speech," *Experimental Psychology*, vol. 57, no. 3, pp. 169–177, 2010.

[22] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proc. ACL*, vol. I, 1992, pp. 517–520.

[23] P. Heeman, A. McMillin, and J. S. Yaruss, "An annotation scheme for complex disfluencies," in *Proc. ICSLP*, 2006.

[24] K. Georgila, N. Wang, and J. Gratch, "Cross-domain speech disfluency detection," in *Proc. Annual SIGdial Meeting on Discourse and Dialogue*, 2010.