Automatic Disfluency Identification in Conversational Speech Using Multiple Knowledge Sources

Yang Liu,[1,3] *Elizabeth Shriberg,*[1,2] *Andreas Stolcke*[1,2]

[1] International Computer Science Institute, Berkeley, CA, USA
[2] SRI International, Menlo Park, CA, USA
[3] School of Electrical and Computer Engineering, Purdue University, Lafayette, IN, USA
{yangl,ees,stolcke}@icsi.berkeley.edu

Abstract

Disfluencies occur frequently in spontaneous speech. Detection and correction of disfluencies can make automatic speech recognition transcripts more readable for human readers, and can aid downstream processing by machine. This work investigates a number of knowledge sources for disfluency detection, including acoustic-prosodic features, a language model (LM) to account for repetition patterns, a part-of-speech (POS) based LM, and rule-based knowledge. Different components are designed for different purposes in the system. Results show that detection of disfluency interruption points is best achieved by a combination of prosodic cues, word-based cues, and POS-based cues. The onset of a disfluency to be removed, in contrast, is best found using knowledge-based rules. Finally, specific disfluency types can be aided by the modeling of word patterns.

1. Introduction

Spontaneous speech differs from written text. One difference is the presence of disfluencies. Accurate identification and clean-up of disfluencies can improve readability and aid performance of downstream language processing modules.

Disfluencies can be broken down into three regions: the reparandum, an optional editing phase[1], and the resumption. Here we study three types of disfluencies:

- repetitions: the speaker repeats some part of the utterance. For example, *I * I like it*.
- revisions (content replacement): the speaker modifies some part of the utterance. For example, *We * I like it*.
- restarts (also called false starts): a speaker abandons an utterance or constituent and then starts over. For example, *It's also * I like it*.

In the examples above, '*' denotes the right edge of the reparandum region and is called the interruption point (IP).

Our goal in this paper is to identify the reparandum region of disfluencies. This represents work carried out for the DARPA EARS project, with the goal of automatic extraction of structural information to enrich automatic transcriptions of speech.

Hindle [1] suggested that an acoustic "edit signal" serves as a cue that fluent speech has been interrupted. Although no evidence for a single such cue has been found, several corpus studies have found that combinations of cues can be used to identify disfluencies with reasonable success [2, 3, 4, 5].

---

[1] The editing phase consists of a spoken cue phrase like filled pauses (such as *uh*), discourse markers (such as *you know, I mean*), or explicit editing terms (such as *oops*).
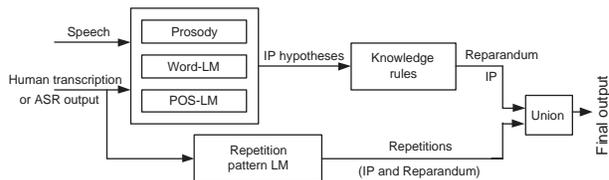

Figure 1: System diagram.

Shriberg and Stolcke [6] proposed a framework for sentence boundary and disfluency detection, which combines a prosody model and a language model. Our study here builds upon that framework, adding more knowledge sources. We investigate additional acoustic-prosodic features, extend the language model to handle repetition patterns, and add a class-based LM. Because speakers are still fluent at the beginning of the reparandum of a disfluency, it is likely that there are no reliable prosodic or language cues at this location. Our approach is thus to first detect the interruption point, and then apply some knowledge-based rules to identify the disfluency starting point. Figure 1 shows the system diagram.

This paper is organized as follows. In Section 2 we describe the acoustic-prosodic features we have investigated. In Section 3, we introduce the language model component, including a POS-based LM and the extended LM that accounts for the repetition patterns. Section 4 shows the experimental results. Conclusions are found in Section 5.

2. Acoustic-Prosodic Features

We extracted acoustic-prosodic features for each inter-word boundary. Similar to prior work [6], two main types of prosodic features, duration and pitch, are extracted from either the forced alignments of speech to human transcriptions, or from speech recognition output. Duration features, such as word duration, pause duration, and phone-level duration, are normalized by overall phone duration statistics and speaker-specific statistics. To obtain F0 features, pitch tracks are extracted from the speech signal and then post-processed to obtain stylized pitch contours [7], from which F0 features are extracted. Examples of F0 features are the distance from the average pitch in the word to the speaker's pitch floor and the difference of the average stylized pitch across a word boundary. Some nonprosodic information is included too, such as speaker gender and turn change.

We also investigated a preliminary set of voice quality measurements, to assess whether they can help identify interruption points. When people stop suddenly, their voice quality can change. Experiments have shown voice quality cues help in de-

tecting word fragments [8]. The following voice quality related features were investigated:

- **Jitter** is a measure of the perturbation in the pitch period [9]. The periodic jitter value is defined as the relative mean absolute second-order difference of the time intervals of the pitch pulse sequence, and is obtained using the Praat tool [10].
- **Spectral tilt** is the overall slope of the spectrum of a speech signal. We use a linear approximation of the spectral envelope to measure spectral tilt.
- **Open Quotient (OQ)** is defined as the ratio of the time in which the vocal folds are open to the total length of the glottal cycle. For the spectral domain, it can be formulated empirically as described in [11].

3. Language Models (LMs)

In order to detect interruption points, we use hidden-event LMs. In a hidden-event LM, each event is represented by an additional non-word token, for example, *I <IP> I think*. The event token <IP> is explicitly represented and included in the vocabulary of the N-gram LM.

3.1. Hidden-Event Word-based LM

The hidden-event word LM models the joint distribution of the event sequence E and the word string W, $P(W, E)$. The word/event LM is trained from the transcriptions, hand-labeled with the events of interest. During testing, a forward-backward algorithm is used to compute the posterior probability $P(E|W)$ and find the most likely event sequence.

3.2. Hidden-Event POS-based LM

We also investigated the effect of a hidden-event LM based on part-of-speech (POS). The idea is to capture syntactically generalized patterns, such as the tendency to repeat prepositions. Heeman and Allen [12] proposed a tightly-coupled approach to finding the best POS sequence and disfluency events together, but experiments were conducted on the TRAINS corpus, which differs from Switchboard conversational speech in that it is far more template-based. As a starting point to incorporate more syntactic information, we used a loosely-coupled model. We trained a POS tagger using the Switchboard Treebank data [13] and used it to tag our training and testing data. Similar to [14], we maintained the identity of some cue words (e.g., filled pauses and discourse markers). Given the tag sequence and the hidden event tokens, we modeled the joint probability of the POS sequence P and the event sequence E. During testing we find the event sequence that maximizes $P(E|P)$ for the given POS string P.

3.3. Repetition Pattern LM

A word-based N-gram LM can only learn certain frequently occurring disfluencies from the training data, and cannot generalize to disfluencies with the same pattern but using different words. For example, in *'I hope to have to have lots of dinner parties'* (with *'to have'* repeated), a regular word-based hidden-event LM fails to detect the IP since the repetition *'to have to have'* does not occur frequently in the training data. Such a failure would also affect the speech recognition task in which the purpose of an LM is to calculate the probability of word strings. To address such issues, we modified the word-based LM to ac-

count for repetitions. Currently we handle only repetitions because they are the most constrained and occur frequently.

For each repetition in the training data, we preserve the cleaned-up utterance, and map the repetition to a pattern. For the example above, the cleaned-up text is *'I hope to have lots of dinner parties'*; the repetition is mapped as follows:

```
            to          have      to          have
  START     ORIG-1      IP        REP-1       END
```

The pattern sequence in the example above is 'START ORIG-1 IP REP-1 END'. The number after '-' in the pattern denotes the position of that event in either the reparandum or the repeat region. The LM is still an N-gram LM, whose counts are obtained from both the cleaned-up text and the counts of such patterns. There is not any lexical context associated with these repetition patterns in the N-gram LM, which is equivalent to allowing such a pattern to occur for any word choice. Note that we model the whole sequence of the repetition pattern (e.g., the IP as well as the reparandum onset) as shown in the pattern example, whereas the regular word-based LM models only one hidden event (the IP).

During testing, for each word boundary we hypothesize repetition events, based on the valid state transitions and whether a word matches a previous word. Each hidden event in a repetition pattern has some properties representing where it occurs in the pattern, from which a possible valid next event can be inferred. During trellis decoding, only valid state transitions are considered. Figure 2 shows the state transitions for repetitions having up to three repeated words.

The probability is calculated in the same way as in a word-based LM for fluent words, until the interruption point. Then, in the repetition, the pattern N-gram probability is used instead of the word-based probability. An advantage of this approach is that it can detect repetitions that have the same pattern as shown in the training data but do not necessarily use the same words; however, an event word-LM can only detect repetitions that have occurred frequently in the training data.

4. Experiments

4.1. Experimental Setup

Experiments were conducted using a portion of the Switchboard-I corpus [15], which consists of 1593 conversations hand labeled for disfluencies [16]. We randomly divided them into a roughly 863K training and 96K test set, with no speaker overlap.

The IP detection task is a two-way classification problem (the top-left block in Figure 1). For each between-word location, a decision of "non-IP" vs. "IP" is made. The IPs of different disfluency types are grouped together into the class "IP", and all other boundaries are grouped into the class "non-IP". We extracted prosodic and voice quality features from both the forced alignments and recognition output, using the SRI LVCSR recognizer [17]. A decision tree was trained from the data to predict event classes using acoustic-prosodic features. Because IPs are relatively rare events, we downsampled our training data to equate the prior probabilities for different classes. This can avoid the problem of highly skewed class distributions and makes the decision tree sensitive to any inherent prosodic features that distinguish the classes. A 4-gram word-based LM and 5-gram POS-based LM were trained from the transcriptions and the annotations of the training set. For the combination of the prosody model and the hidden-event word LM, we used an HMM-based integration approach [6]. When
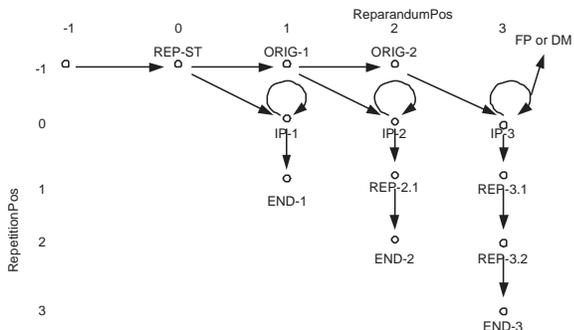
Figure 2: The valid state transitions for repetitions of up to 3 words. X-Y axes represent the position in the reparandum and repeat region, with events denoted by ORIG- and REP- respectively. In ORIG-n, *n* means the position of a word in the reparandum; in REP-m.n, *m* is the total number of repeated words and *n* represents the position of the event in the repeat region. Optional filled pauses (FP) or discourse markers (DM) are allowed after the IP in the transition.

combining the three models (the prosody model, the word-based LM, and the POS-based LM), we interpolated the posterior probabilities from all three models.

In the system diagram (Figure 1), the repetition pattern LM detects the existing repetitions. The IPs and the onset of the reparandum are among the outputs of this module. After detecting interruption points, we use some rule-based knowledge to identify the start point of the disfluencies. The final output for the system is the union of the decisions from the rule-based post-processing module and the repetition pattern identifier.

4.2. Experimental Results

4.2.1. IP Detection

We report experimental results using classification accuracy, recall, and precision. Because there is a tradeoff between recall and precision, we optimize the parameters for model combination on the overall accuracy. The top rows of Table 1 show the results on the downsampled test data using a prosody model alone. The bottom rows are the results on non-downsampled test data using the word-based LM, the POS-based LM, the prosody model, and their combination. All results were obtained from testing on human transcriptions in order to avoid the effect of word errors in recognition outputs when comparing the performance of different models. Note that in this test condition, we did not use any word fragment information (which, when present, always signals the existence of disfluencies) for a fair comparison with testing on ASR output, since current ASR systems do not provide word fragment information.

The results in Table 1 show that the prosody model alone yields a much better accuracy than chance performance on downsampled data. This suggests that there exist some acoustic-prosodic cues at the interruption points. However, we need to be careful when interpreting this result. Because the non-downsampled test data is unbalanced between the two classes, using only the posterior probability from the decision tree on the test set will generate many false alarms. In order to take into account the prior probability of an IP, the final decision on the non-downsampled data is a combination of the prior probability of each class (IP vs. non-IP) and the posterior probability given by the decision tree. We found that using such an approach, the prosody model alone yields only chance performance on the non-downsampled data. This is because the pos-

Table 1: IP detection results using the human transcriptions. Chance performance on the non-downsampled data is obtained by hypothesizing each word boundary as a 'non-IP' event.

		Recall	Precision	Overall Accuracy
Downsampled	Chance	-	-	50
	Prosody	75.81	77.26	76.75
Non-downsampled	Chance	0	-	96.62
	Prosody	0	-	96.62
	Word-LM	55.47	79.33	98.01
	POS-LM	36.73	65.75	97.22
	Word-LM +Prosody	58.27	78.37	98.05
	Word-LM + Prosody + POS-LM	56.76	81.25	98.10

terior probability generated by the prosody model is not very high, implying the prosodic features are not sufficiently reliable to overtake the low prior probability of an 'IP' event. Hence the final decision is always 'non-IP'. Experiments in [18] have shown that useful prosodic cues exist at the interruption points, but the performance of the prosody model was not investigated on non-downsampled data in that research.

An analysis of the results shows that most of the interruption points correctly detected by LMs are repetitions. It is more difficult to capture the properties of the revisions and restarts by a simple N-gram word model. The word-based LM alone outperforms a POS-based LM alone, indicating the importance of specific lexical information.

Interestingly, even though when combined with a word-based LM, neither the prosody model nor the POS-based LM yield significant improvement over the single word LM[2], the combination of the three models achieves the best performance, virtually a 4.5% relative reduction of total classification error rate compared to using a word-based LM (significance binomial test shows at the level of 0.02). This suggests that every knowledge source provides some information and that their combination yields improvement overall.

Table 2: Comparisons of IP detection on human transcriptions (Ref) and ASR output.

		Chance	Accuracy
Downsampled Prosody	Ref	50	76.75
	ASR	50	72.61
Non-downsampled Word-LM	Ref	96.62	98.01
	ASR	96.70	97.05

Table 2 shows results when testing on recognition output compared to human transcriptions. Note that in Table 2 we do not show all model combinations; results are similar to the patterns obtained using human transcriptions. When testing on recognition output, we observe degradations in performance. Word errors affect the robustness of both language models and the prosody model, with more effect on LMs. As shown in Table 2, there is less degradation to the prosody model on downsampled data than to the LM on non-downsampled data[3]. This

---

[2]We have conducted experiments and found that the prosody model contributes differently for different types of disfluencies. For example, combining with the prosody model can reduce the classification error (statistically significant) in the task of repetition interruption point detection. However, our task here groups all IPs into one class.

[3]Because the prosody model alone performs at chance, we cannot observe the effect of recognition output on the prosody model using

suggests that LMs are more dependent on word identity and thus are more affected in face of incorrect words than is the prosody model.

4.2.2. Repetition Pattern Detection

We tested the repetition pattern LM on the human transcriptions for repetition detection. We also used this LM to calculate the perplexity of the test set. The results are shown in Table 3. Although this LM can detect the IP and onset of the reparandum together, we report only IP detection results in order to compare results with those for the word-based LM. We found that more repetitions are identified using this pattern LM than when using the word-based LM. Spot-checking of the reference annotation shows some errors in the reference, suggesting that the results of using the repetition pattern LM may be underestimated. Note that a repetition detection model that takes into account the word fragment information would further improve the accuracy of results shown in the table. This repetition pattern LM requires strict word matching, therefore, it also suffers from ASR errors.

The perplexity generated by the pattern LM is reduced compared to that of the word-only LM. In repetitions, the word-only LM uses the word-based probability, whereas the pattern LM calculates only the pattern event probability (which usually has a higher probability than the word sequence). Whether we can use such a LM for rescoring lattices or N-best lists is currently under investigation.

Table 3: Repetition IP detection accuracy and perplexity on the test set using a repetition pattern LM and a word-based LM. The hidden-event in the word-based LM is '<IP>'.

	Recall	Precision	Perplexity
Repetition Pattern LM	80.67	70.09	110.95
Word-based LM	67.77	77.29	125.96

4.2.3. Finding the Onset of Disfluencies

After IPs are detected, as described earlier, we use rule-based knowledge to find the onset of the reparandum. For example, a repeated word across an IP helps delimit the reparandum (e.g., in "*I want to leave on Monday * on Sunday*", the reparandum starts from the first "*on*"). We use a boundary based classification accuracy as our performance measure for the disfluency starting point. We obtained a recall rate of 61.45% and precision of 68.64%, compared to a recall of 46.41% and precision of 75.86% using a hidden-event LM alone to find the start of disfluencies. Better results would be obtained using more accurate IP hypotheses. Ultimately our goal is to jointly model these different knowledge sources.

5. Conclusions

We have described a disfluency detection system based on acoustic-prosodic features, as well as word-based, POS-based, and repetition pattern based LMs. For IP detection, results show that a prosody model alone performs much better than chance on downsampled test data, but only performs at chance on the non-downsampled data. A word-based hidden-event LM alone outperforms both the prosody model and the POS-based LM alone. When combined with a word-based LM, neither the prosody model nor POS-based LM yields significant improvement over the single word-based LM. Interestingly however,

non-downsampled data.

the combination of the three models achieves the best performance. This suggests that each knowledge source contributes differently to the combined performance. We also find that taking repetition patterns into account in the LM can help detect repetitions and reduce perplexity. Finally, recognition errors degrade classification performance. The incorrect recognized words have a more negative effect on LMs than on the prosody model, because of the stronger dependence of LMs on correct word identity.

7. References

[1] D. Hindle, "Deterministic parsing of syntactic nonfluencies," in *Proc. ACL*, 1983, pp. 123–128.

[2] J. Bear, J. Dowding, and E. Shriberg, "Integrating multiple knowledge sources for detecting and correction of repairs in human-computer dialog.," in *Proc. ACL*, 1992, pp. 56–63.

[3] R. J. Lickley, "Juncture cues to disfluency," in *Proc. ICSLP*, 1996.

[4] C. Nakatani and J. Hirschberg, "A corpus-based study of repair cues in spontaneous speech," *JASA*, pp. 1603–1616, 1994.

[5] E. Shriberg, "Phonetic consequences of speech disfluency," in *Proc. ICPhS*, 1999, pp. 619–622.

[6] E. Shriberg and A. Stolcke, "Prosody modeling for automatic speech recognition and understanding," in *Proc. Workshop on Mathematical Foundations of Natural Language Modeling*, 2002.

[7] K. Sonmez, E. Shriberg, L. Heck, and M. Weintraub, "Modeling dynamic prosodic variation for speaker verification," in *Proc. ICSLP*, 1998, pp. 3189–3192.

[8] Y. Liu, "Word fragment identification using acoustic-prosodic features in conversational speech," in *HLT-NAACL student research workshop*, 2003, pp. 37–42.

[9] A. E. Rosenberg, "The effect of glottal pulse shape on the quality of natural vowels," *JASA*, vol. 49, pp. 583–590, 1970.

[10] P. Boersma and D. Wennik, "Praat, a system for doing phonetics by computer," http://www.praat.org, 1996.

[11] G. Fant, "The voice source in connected speech," *Speech Communication*, vol. 22, pp. 125–139, 1997.

[12] P. Heeman and J. Allen, "Speech repairs, intonational phrases and discourse markers: Modeling speakers' utterances in spoken dialogue," *Computational Linguistics*, 1999.

[13] LDC, "http://www.ldc.upenn.edu," .

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

[15] J. Godfrey, E. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proc. ICASSP*, 1992, pp. 517–520.

[16] M. Meteer, A. Taylor, R. MacIntyre, and R. Iver, "Dysfluency annotation stylebook for the switchboard corpus," Distributed by LDC, 1995.

[17] A. Stolcke, H. Bratt, and et al., "The SRI March 2000 Hub-5 conversational speech transcription system," in *Proc. NIST Speech Transcription Workshop*, College Park, MD, May 2000.

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