Comparing HMM, Maximum Entropy, and Conditional Random Fields for Disfluency Detection

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

[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

Automatic detection of disfluencies in spoken language is important for making speech recognition output more readable, and for aiding downstream language processing modules. We compare a generative hidden Markov model (HMM)-based approach and two conditional models — a maximum entropy (Maxent) model and a conditional random field (CRF) — for detecting disfluencies in speech. The conditional modeling approaches provide a more principled way to model correlated features. In particular, the CRF approach directly detects the reparandum regions, and thus avoids the use of ad-hoc heuristic rules. We evaluate performance of these three models across two different corpora (conversational speech and broadcast news) and for two types of transcriptions (human transcriptions and recognition output). Overall we find that that the conditional modeling approaches (Maxent and CRF) provide benefit over the HMM approach. Effects of speaking style, word recognition errors, and future directions are also discussed.

1. Introduction

Automatic detection and correction of disfluencies can aid both human readability and downstream automatic processing (e.g., parsing, machine translation, information extraction). The term 'disfluency' has different meanings for different researchers, and can include hesitation pauses, filled pauses (e.g., *um*, *uh*), and discourse markers (e.g., *you know*, *so*, *like*). Here we use this term to refer to cases for which there is a surface form speech repair (also called an edit disfluency), which follows the template below [1]:

(reparandum) * ⟨editing term⟩ [correction]

The location denoted by '*' is an interruption point (IP), or the point in time at which the speaker breaks off from the original utterance, prior to any corrections (including repeats).

In previous work on automatic disfluency detection, a number of efforts have focused on textual information only [2, 3, 4, 5, 6, 7]; others have attempted to find effective acoustic and prosodic cues [8, 9, 10, 11]; and still others have combined textual and prosodic information [12, 13, 14]. Most such studies have been conducted on human transcriptions. Although investigation of human transcriptions can shed light on potential features and useful models, our final goal is to enrich speech recognition results, which contain word errors that affect the modeling approaches. Therefore, it is important to examine the impact of speech recognition on the automatic disfluency detection task. In addition, many prior studies on disfluency detection rely on the availability of sentence boundary information. This information is generally unavailable given speech recognition output, and so must be inferred automatically during processing, making disfluency detection even more difficult.

For model training, some of the previous methods require the annotation of both the reparandum and the corrections of an edit disfluency [5, 6, 15] to make use of the correspondence between them. In our data set, described in Section 3, as well as in many other corpora that do not contain full disfluency markup, information about the correction region is unavailable. We are therefore more interested in algorithms that do not require the use of this information in training.

We chose to explore three modeling approaches (a hidden Markov model, a maximum entropy model, and a conditional random field) for edit disfluency detection. The approaches themselves have been used in other speech and language processing tasks. For example, they have been used for sentence boundary detection [16], in which the conditional modeling approaches have shown advantages at modeling overlapping textual features.

The challenge and novel contribution in the present study was to develop both the architectures and the features for these models, as applied to the specifics of the edit disfluency task. In addition, we ask how models are affected by speech recognition errors, as well as how they are affected by different speaking styles.

The remainder of the paper is organized as follows. In Section 2, we describe the different approaches used for automatic disfluency detection. Experimental results are shown and discussed in Section 3. A summary and future work appear in Section 4.

2. Methods

2.1. Knowledge Sources

To detect edit disfluencies, we use different knowledge sources involving both textual information and prosodic cues. Textual information obtained from transcriptions (either human transcriptions or automatic speech recognition output) is no doubt very important; in many cases, people have little difficulty interpreting a disfluent utterance from the word transcription alone. In our edit disfluency detection system, textual information is represented by word identity, the co-occurrence of two or more words, part-of-speech tags or semantic classes, and information about word repetitions. However, in some cases, the use of textual information alone may not completely disambiguate structural events.

Prosody provides information that is complementary to the word sequence, and may also be more robust in the face of

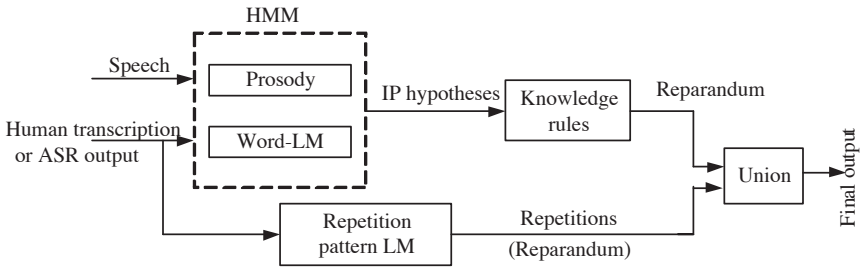

Figure 1: Edit word detection using HMM for IP detection.

word errors in speech recognition output. Past research results suggest that speakers use prosody to impose structure on both spontaneous and read speech. The prosodic features we use are associated with interword boundaries and can be automatically extracted from the word and phonetic alignments of the word sequence. We compute prosodic features including duration, fundamental frequency (F0), energy, and pause [17]. A decision tree classifier is used as the prosody model that generates the posterior probability of an event given the prosodic features at an interword boundary for all classifier frameworks.

2.2. Modeling Approaches

2.2.1. Hidden Markov Model (HMM)

Since speakers are still fluent at the starting point of an edit disfluency, it is not likely that there are acoustic or lexical cues at that location. Cues are often present, however, at the point at which the speaker interrupts his or her speech. Therefore, our HMM first uses prosodic and lexical features to detect the IP and then uses knowledge-based rules to locate the corresponding reparandum onset for the hypothesized IPs. Figure 1 shows the diagram for the HMM-based approach.

The top left box in Figure 1, shown with a dashed line, represents a two-way classification model for IP detection. In this HMM framework for IP detection, the transition probabilities are modeled by an N-gram language model (LM) of the word/tag sequences. Additional prosodic features are modeled as observation likelihoods attached to the N-gram states of the HMM [16]. We use a decision tree prosody model that is trained from the downsampled training set because of the highly imbalanced distribution of IP and nonIP events. The most likely event at each location is then found using forward-backward decoding.

A word-based N-gram LM can learn only certain frequently occurring disfluencies from the training data and tends not to generalize well to other related disfluencies involving different words. To address this issue, we have expanded the word-based LM to account for repetitions [12]. The "knowledge rules" box in Figure 1 applies heuristic knowledge to determine the extent of the reparandum in a disfluency after the IP is detected. Linguistic studies suggest that people tend to start from the beginning of a constituent in repetitions or revisions (e.g., repeating the function words). For example, a revision disfluency may be "*a red a blue car*". If the IP is correctly hypothesized at the interword boundary, that is "*a red* $\langle IP \rangle$ *a blue car*", then we can go backward to find whether the same word as the word after the IP ("*a*" in this example) has occurred before the IP and thus determine the potential onset of an edit disfluency.

2.2.2. Maximum Entropy (Maxent)

The second approach we used is a Maxent classifier for IP detection. A Maxent classifier estimates the posterior probabilities directly, compared to the generative modeling approach of the HMM. Such a model takes the exponential form

$$P(E_i|W_i,F_i) = \frac{1}{Z(W_i,F_i)} e^{\sum_k \lambda_k g_k(E_i,W_i,F_i)} \quad (1)$$

where Z is a normalization term and i is the index for the word boundary location. The indicator functions $g_k(E_i,W_i,F_i)$ correspond to features defined over events, words, and prosody. The parameters in Maxent are chosen to maximize the conditional likelihood $\prod_i P(E_i|W_i,F_i)$ over the training data, and thus better match the classification accuracy metric.

In the Maxent approach, we first apply a classifier with the following classes: SU (meaning a sentence boundary), IP, or NULL. Then, similarly to the HMM, heuristic rules are used to determine the onset of the reparandum. Note that the classes are different from those used in the HMM IP detection module, which involves a two-way classification task, since we observed in previous experiments that using two classes outperforms using three target classes in the HMM. However, for the conditional Maxent approach, we expect that the model may be better at jointly modeling SUs and IPs and learning how to distinguish between them. For example, if "*that is great. that is great*" has occurred in the training set, then the model will learn that these are two SUs, rather than an edit disfluency, even though the word sequence is repeated. In the repetition detection module in the HMM, we need to predefine some cue words that should be SUs and so would not be considered edit disfluencies (such as "*uhhuh uhhuh*"). The probabilistic Maxent model, on the other hand, is able to learn these kinds of cue words from the training set and thus model them more elegantly.

The features used in the Maxent model for the SU/IP/NULL detection task are as follows:

- All the textual features and prosodic information used for SU detection, such as N-grams of words and various tags for the words. Details can be found in [16].
- Repetition information. At each word boundary, this feature represents whether there is a repeated word sequence (as many as three words) that ends at that point, with optional filler words allowed starting from that point.
- Fragment information. This feature represents whether a word is a fragment. Only in the reference transcription condition can this feature be triggered. In the speech recognition output condition, no word fragment information is available.

- Filler words. This feature represents whether there is a predefined filler phrase after a word boundary given a list of cue words.
- Prosody model posterior probabilities. This is obtained from the same prosody model as used in the HMM IP detection module. The posterior probabilities are discretized by cumulative binning [16].

2.2.3. *Conditional Random Fields (CRF)*

A CRF is a random field that is conditioned on an observation sequence [18]. It is essentially a Maxent model over the entire sequence. It differs from the Maxent above in that it models the sequence information, whereas the Maxent model makes a decision for each state independently of the other states.

The CRF approach we have designed for edit word detection finds the entire region of the reparandum, similarly to the named entity recognition task [19]. Each word has an associated tag, representing whether or not it is an edit word. The classes used in the CRF edit word detection approach are: the beginning of an edit (B-E); inside of an edit (I-E); each of which has a possible IP associated with it (B-E+IP or I-E+IP), and outside of an edit (O), resulting in five states. The following is an example of a transcription excerpt together with class tags used for the CRF edit word detection model:

```
I      I   work   uh  i'm an analyst
B-E+IP I-E I-E+IP O   O   O   O

and it   got    it got real rough
O   B-E  I-E+IP  O  O   O    O
```

Note that IPs are included in the target class when using the CRF for edit detection in order to identify the internal IPs inside complex edit disfluencies.[1] For example, "*I I work*" in the above example is the reparandum in a complex edit disfluency, with an internal IP after the first "*I*".

Features used in the CRF method are the N-grams of words and their part-of-speech tags, speaker turn change, and all the features used by the Maxent IP detection model that are not used for SU detection. Note that we used fewer features[2] in the CRF approach than in the Maxent model considering that CRF training is computationally expensive.

3. Experiments

3.1. Experimental Setup

We used the data from the most recent RT-04F NIST evaluation [20] for the DARPA EARS program. The data has been annotated by LDC according to guidelines [1] with information about the reparandum of an edit disfluency, along with sentence boundaries (called "SUs") and filler words. Evaluation is conducted using two different corpora: conversational telephone speech (CTS) and broadcast news speech (BN). The speaking styles in these two corpora is very different. The average length of sentences is much shorter in CTS. Disfluencies are more frequent in CTS than in BN. We evaluated using either human transcriptions (REF) or speech recognition output (STT).

[1] For simple edit disfluencies, the end of an edit disfluency signals an IP.

[2] We preserve the features that we expect are more relevant to the edit disfluency detection task. Some features used in the Maxent model were originally designed for the SU detection task [16].

Table 1: Data description for CTS and BN used in the RT-04 NIST evaluation, including data size (number of words), the word error rate of the recognizer, and the percentage of the edit words.

	BN	CTS
Training	353k	484k
Test	45k	35k
STT WER (%)	11.7	14.9
Edit word (%)	1.8	7.5

Table 2: Results (NIST error rate in %) for edit word and IP detection, using the HMM, Maxent, and CRF approaches on the reference and recognition output conditions of CTS data.

CTS				
Approaches	Edit word		Edit IP	
	REF	STT	REF	STT
HMM	54.33	85.32	33.21	73.66
Maxent	55.89	87.86	34.11	73.72
CRF	50.07	80.41	34.80	72.61

In both cases, we extracted prosodic features from speech. Table 1 shows the data size, the word error rate (WER) of the recognizer in the test set, and the frequency of edit disfluencies in the two corpora.

Performance is measured based on the number of the incorrectly identified words per reference edit word or IP in the NIST scoring tool [20]. Note the performance metric is measured as per reference edit word; hence the denominator is generally small, making the error rate look high in some cases. When recognition output words are used, they usually do not align perfectly with those in the reference transcriptions. In this case, an alignment that minimizes the WER is used. After alignment, the hypothesized edits are mapped to the reference edits using the word alignment information, and then unmatched edits are counted.

3.2. Results and Discussion

Table 2 shows the results for the different models on both edit word and IP detection using the CTS corpus. For the reference condition, the CRF is better at finding edit words, but slightly poorer at IP detection compared with the HMM or Maxent methods. This ties into how the models are trained: the HMM and Maxent are trained to detect IPs, but the heuristic rules used may not find the correct onset for the reparandum; the CRF, on the other hand, is trained to jointly detect the edit words and IPs and thus may not be as well trained for IP detection. However, on the STT condition, we observe that the CRF approach outperforms both the Maxent and HMM methods for both the edit word and edit IP tasks, suggesting that the CRF degrades less for the edit IP detection task as a result of word errors. This probably occurs because edit word and IP detection are mutually beneficial. The poorer performance of the Maxent model compared with the HMM could be attributed to the use of discretized posterior probabilities from the prosody model, which reduces the impact of prosodic information.

Table 3 shows the results for BN edit word and IP detection, using the HMM and the Maxent approaches. A CRF was not used for the BN data because disfluencies are rare in BN,

Table 3: Results (NIST error rate in %) for edit word and IP detection in BN, using the HMM and Maxent approaches.

BN				
Approaches	Edit word		Edit IP	
	REF	STT	REF	STT
HMM	45.96	93.20	34.30	93.20
Maxent	43.00	89.86	30.89	87.54

and CRF training is very computationally expensive. The Maxent approach yields better results for both edit word and IP detection than the HMM, in contrast to the CTS corpus. Performance degrades severely in the STT condition compared with reference transcriptions, with a larger error rate increase than observed for the CTS edit word and IP detection tasks.

We observe from Tables 2 and 3 that the error rate for edit word detection in BN is not worse than in CTS for the reference condition, even though the percentage of the edit words is much smaller in BN than in CTS, which significantly reduces the denominator used in the performance measure. This suggests that to some extent for the reference condition, edit word detection is a relatively easier task in BN than in CTS. This makes sense because of the different speaking styles found in the two corpora. Also note that an important feature for detection of edit words and IPs is the occurrence of word fragments, which are provided in the reference condition but are unavailable in the STT condition. The severe performance degradation in the STT condition may result from the unavailability of word fragment information, as well as from the word errors in edit disfluency regions.

4. Conclusions

Automatic disfluency detection is important for improving readability of speech recognition output and for making speech easier to process for downstream language modules. We have applied three modeling approaches to automatic disfluency detection using two corpora (conversational speech and broadcast news) and two types of transcriptions (human transcripts and recognition output). The approaches we have examined here do not require the annotation of the correction region in disfluencies for model training.

Comparing the three modeling approaches, we find that the conditional modeling approaches (Maxent and CRF) provide an elegant and successful way to model various and potentially correlated features. The CRF approach, as implemented here, detects the reparandum and IP; it thus avoids the use of ad-hoc rules used in the HMM and Maxent methods. In terms of effects of transcripts and corpora, we find that performance degrades substantially due to word errors in recognition output. We also observe differences between corpora related to both quantitative and qualitative (type) differences in disfluencies for conversational speech versus broadcast news.

Much work still remains to improve system performance for edit detection. We believe that syntactic information is an important knowledge source for improving results. Furthermore, a more effective use of prosodic features should contribute to future progress. Finally, recent work includes the study of the impact of disfluencies on subsequent language processing modules, such as parsers.

5. Acknowledgments

This work has been supported by DARPA under contract MDA972-02-C-0038, and NSF-STIMULATE under IRI-9619921, NSF KDI BCS-9980054, and ARDA under contract MDA904-03-C-1788. Distribution is unlimited. Any opinions expressed in this paper are those of the authors and do not reflect the funding agencies. Part of the work was carried out while the last author was on leave from Purdue University and at NSF.

6. References

[1] S. Strassel, *Simple Metadata Annotation Specification V6.2*, Linguistic Data Consortium, 2004.

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

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

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

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

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

[7] M. Honal and T. Schultz, "Corrections of disfluencies in spontaneous speech using a noisy-channel approach," in *Proc. of Eurospeech*, 2003.

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

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

[10] G. Savova and J. Bachenko, "Prosodic features of four types of disfluencies," in *Proc. of DiSS*, 2003, pp. 91–94.

[11] R. Kompe, *Prosody in Speech Understanding System*, Springer-Verlag, 1996.

[12] Y. Liu, E. Shriberg, and A. Stolcke, "Automatic disfluency identification in conversational speech using multiple knowledge sources," in *Proc. of Eurospeech*, 2003, pp. 957–960.

[13] M. Snover, B. Dorr, and R. Schwartz, "A lexically-driven algorithm for disfluency detection," in *Proc. of HLT/NAACL*, 2004.

[14] 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.

[15] M. Lease, M. Johnson, and E. Charniak, "Parsing and its applications for conversational speech," in *Proc. of ICASSP*, 2005.

[16] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper, "Comparing and combining generative and posterior probability models: Some advances in sentence boundary detection in speech," in *Proc. of EMNLP*, 2004.

[17] E. Shriberg, A. Stolcke, D. Hakkani-Tur, and G. Tur, "Prosody-based automatic segmentation of speech into sentences and topics," *Speech Communication*, pp. 127–154, 2000.

[18] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random field: Probabilistic models for segmenting and labeling sequence data," in *Proc. of ICML*, 2001, pp. 282–289.

[19] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields," in *Proc. of CoNLL*, 2003.

[20] National Institute of Standards and Technology, "NIST RT Fall 2004 Evaluation," http://www.nist.gov/speech/tests/rt/rt2004/fall/, 2004.