Detecting Laughter and Filled Pauses Using Syllable-based Features

Gouzhen An[1], *David Guy Brizan*[1], *Andrew Rosenberg*[1,2]

[1] Department of Computer Science, CUNY Graduate Center, USA
[2] Department of Computer Science, Queens College (CUNY), USA
{ gan@gc.cuny.edu, dbrizan@gc.cuny.edu, andrew@qc.cuny.edu }

Abstract

Identifying laughter and filled pauses is important to understanding spontaneous human speech. These are two common vocal expressions that are non-lexical and incredibly communicative. In this paper, we use a two-tiered system for identifying laughter and filled pauses. We first generate frame level hypotheses and subsequently rescore these based on features derived from acoustic syllable segmentation. Using Interspeech 2013 ComParE challenge corpus, SVC, we find that these rescoring experiments and inclusion of syllable based acoustic/prosodic features allow for the detection of laughter and filled pauses by at 89.3% UAAUC on the development set, an improvement of 1.7% over the challenge baseline.

Index Terms: laughter detection, filled pause detection, prosodic analysis

Introduction

Identifying laughter and filled pauses is important to understanding spontaneous human speech. These are two common vocal expressions that are non-lexical and incredibly communicative. Both laughter and filled pauses can occur at virtually any point in the speech stream, posing particular difficulty for language modeling.

Laughter can be used to indicate amusement, happiness, discomfort, scorn or embarrassment. The role of filled pauses ("um", "er", "uh", etc.) has received a significant amount of attention, but it has been hypothesized that these are used to hold the floor while planning to allow the necessary time to retrieve a concept or construct a syntactically or semantically complicated utterance. Others have hypothesized that fillers prime a listener to receive unexpected, or discourse new information [1].

Interactive Voice Response (IVR) systems can make use of laughter and filler detection. Primarily, an IVR would benefit by not mistaking laughter or fillers as lexical content. Additionally, a system that can infer what the speaker is trying to communicate through these phenomena will have the ability to provide a more natural, and probably more successful dialog with a user. For example, hesitation and laughter may be evidence that the IVR system has confused the user or made an unexpectedly funny misrecognition of the speaker's input. In a tutoring system, hesitation may be an indicator of a lack of certainty in the student [2]. Moreover, this work can be applied in nativeness and fluency assessment. Filled pauses and other disfluencies are significantly higher in non-native speech than in native speech [1]. We describe some relevant previous work on the recognition of laughter and filled-pauses in Section 2.

In this paper we describe a two-stage approach to detecting laughter and filler. This work describes a submission to the Interspeech 2013 ComParE Social Signals Sub-challenge [3]. In this task, each 10ms of the stimulus is annotated as **laughter**, **filler**, or **garbage**, which includes speech, silence and other non-speech noise. Following the baseline classification, we first use 10ms frame-level features to detect these laughter and filler events using SVM classification. However, frame-based detection does not effectively incorporate 1) prosodic content, 2) durational properties of laughter and filler events or 3) transition likelihoods from one event to another. We use an acoustic-based pseudosyllabification approach to segment each utterance into syllable-like regions. This segmentation information is incorporated into the second stage of our detection process by 1) defining a region-of-analysis from which we extract additional acoustic/prosodic information, 2) determining a segment based on duration and 3) re-scoring segment-internal frames differently from frames close to segment boundaries. In Section 3, we describe a number of rescoring experiments, exploring different approaches to leverage segmentation information.

We find that incorporating features drawn from pseudosyllable segmentation and rescoring frame-level hypotheses are able to improve the detection of laughter and filler by 1.7% UAAUC over the frame-level baseline.

Related Work

There has been a range of work on the identification and acoustic characterization of laughter. Because laughter is a social signal, most previous work detects laughter in dialog speech with a particular focus on multiparticipant meetings [4, 5]. While there have been some hypotheses about specific acoustic realizations of laughter, the consensus is that laughter is a heterogeneous phenomenon and highly variable, even within a single speaker [6]. Trouvain describes a number of different stereotypes about laughter; that it can be "snort"-like, "grunt"-like, "song"-like, and it may be vocalized or not, but even these variables are insufficient to describe the "large repertoire of laughter variances" [6]. Tanaka and Campbell [7] define four (or five) types of laughter in Japanese speech: mirthful, polite, embarrassed, derisive and other. Though even in this taxonomy, they find it difficult to distinguish embarrassed and polite laughter.

The acoustic analysis of laughter focuses on models with features drawn from F_0 and spectral (MFCC and PLP) features. These features are included in the Interspeech 2013 ComParE challenge baseline feature set [3]. In addition, Schuller et al. explore laughter detection by finding "coarse repetition of vowel sounds" which are captured by high-energy pulses every 200-250 milliseconds [8].

The detection of filled pauses is typically accomplished in automatic speech recognition by including "um", "er", and "uh" as lexical items. There has, however, been some work that analyzes the acoustic properties of filled pauses. For example, in [5] the authors identify a "nearly constant fundamental frequency (F_0) and minimal spectral envelope deformation." Goto et al. similarly are able to characterize

filled pauses from lexical syllables in Japanese by increased duration, sustained pitch and stable harmonics [9].

Method

In this section, we outline our approach in two types of experiments: 1) Rescoring Experiments (cf. Section 3.2), where frame level hypotheses are rescored at different thresholds based on their surrounding context, and 2) Syllable-based Feature Experiments (cf. Section 3.3), which incorporate longer-range acoustic/prosodic features based on acoustic syllabification (cf. Section 3.1) into the decision making process. Each set of experiments uses the same framework; as a part of the Interspeech 2013 ComParE challenge, all experiments are performed using the challenge train, development and test sets of the SVC Corpus [3]. This corpus is made up of task-based telephone conversations between strangers. In the reported experiments, we train our models on the training material and evaluate the performance on the development material.

The initial hypotheses used in the rescoring experiments are the baseline challenge results. These are generated using the baseline feature set and SMO (SVM) classification with C=0.1 and C=0.001. The performance of the baseline classifier is better with C=0.1, but we found that training was considerably faster with C=0.001. Therefore we tuned our rescoring experiments with C=0.001 and applied the best performing configurations with C=0.1.

1.1. Pseudosyllabification Approach

An acoustic segmentation of the input material forms the basis of our rescoring experiments. For this segmentation, we use a pseudosyllabification algorithm described by Villing et al. [10] and implemented in AuToBI [11]. This algorithm operates by assuming that amplitude peaks are associated with syllable nuclei, and valleys are syllable boundaries. The waveform is passed through an equal loudness filter, and a low-pass filter. Onset velocities are determined by the maxima of the filtered envelope slopes. These onsets are used as candidate syllable boundaries. Boundaries are then selected based on a scoring function incorporating the velocity of the onset and spectral content at the candidate vowel peak. An additional temporal filter is applied to the candidate boundaries to suppress weak boundaries within 100ms of strong boundaries. This also serves to eliminate spurious boundaries at the start and end of an utterance. The implementation used in this paper is available as part of AuToBI and can be found at `http://speech.cs.qc.cuny.edu/autobi`

1.2. Rescoring Experiments

The baseline feature set includes 141 acoustic features centered on each 10ms frame and 8 neighboring (4 preceding and 4 following) frames. These features are used to train an SVM classifier with C=0.1 and a significantly (5%) undersampled majority class, **garbage**. When inspecting the results of this approach on the development set, we observe that many of the frames have labels different from the others in the same vicinity. Moreover, this typically happens when the model has predicted a label with low confidence. This results in occasional predictions of extremely short regions, sometimes just one or two frames, of laughter or filler. One example of this is shown in Figure 1. Laughter and filled pauses last longer than 10-20ms; we know this intuitively, and have confirmed through inspection of the corpus annotations. Our goal in these rescoring experiments is to smooth the confidence scores of the first pass baseline classifier, to 1) improve the overall performance by taking broader context into account and 2) eliminate these short spurious predictions.

```
'S1393.wav',188,garbage,0.700,0.065,0.235
'S1393.wav',189,garbage,0.575,0.093,0.332
'S1393.wav',190,garbage,0.474,0.126,0.400
'S1393.wav',191,filler,0.425,0.113,0.462
'S1393.wav',192,garbage,0.478,0.091,0.430
'S1393.wav',193,garbage,0.534,0.080,0.385
'S1393.wav',194,garbage,0.692,0.069,0.239
'S1393.wav',195,garbage,0.783,0.054,0.163
'S1393.wav',196,garbage,0.792,0.072,0.136
```

Figure 1: *Output of baseline classifier*

Our analysis of the output of the initial hypothesis suggests that the classifier produces erroneous results which can be detected by observing the average (mean) prediction for the surrounding frames or, failing that, by observing the majority prediction of the surrounding frames. This analysis drives our rescoring techniques.

In the rescoring experiment, we first use AuToBI to segment the source material into pseudo-syllable regions. Our hypothesis here is that within one syllable there should be at most one filler or laughter event and no other material. For each prediction within each segment, we apply a rescoring algorithm to change minority labels which are different from others in the same region. We experiment with three different rescoring approaches:

1. Average Segment Threshold
2. Majority Class Threshold
3. Label Exchange Threshold

In each approach, we establish a confidence score for each frame label in the segment, applying this label to the contained frames.

1) Average Segment Threshold. In this approach we first calculate the unweighted average confidence score for each label in each segment. We then compare this aggregated score to class based thresholds. If the score for a class, **garbage, laughter or filler,** exceeds the threshold, we assign this confidence score to all frames in the segment. If none exceeds the threshold we fall through to the Majority Class Threshold. We notice that the garbage is over-represented in the material, so we establish higher thresholds for garbage than for the other two classes. That is, in order for this algorithm to predict **garbage** there must be more evidence for that class than for one of the non-garbage classes. We experiment with three thresholds for garbage vs. non-garbage: [0.9, 0.7], [0.8, 0.6] and [0.7, 0.5]. Based on tuning experiments we find that [0.9, 0.7] represent the most effective threshold.

2) Majority Class Threshold. Here we calculate the fraction of labels assigned to each class in a segment by counting the number of labels per class and dividing by the total number of labels in the segment. Effectively we apply a max function to each distribution of confidence scores prior to calculating the average. When this fraction exceeds our threshold, we establish this class as our *segment label*. If we are unable to determine a segment label because none exceeds our threshold, we fall through to the Label Exchange Threshold. Again, because of the overrepresentation of garbage, we establish higher thresholds for that class than for laughter or for filler classes. We experiment with the same thresholds as above: [0.9, 0.7], [0.8, 0.6] and [0.7, 0.5], finally settling on the same threshold: [0.9, 0.7].

3) Label Exchange Threshold. If the classifier is not confident enough to consistently label the frames within a

segment at rates higher than the thresholds for Average Segment and Majority Class Thresholding, we assign the a *segment label* with the highest average score in the segment, identical to the Average Segment Threshold algorithm, but established with no thresholds.

Once a *segment label* has been established, we then compare the predicted label for an individual frame to the label for the segment. We change the frame's label if it falls below a certain threshold. In this way, we preserve the labels for frames for which the first pass prediction has high confidence, but assign the segment label to the lower confidence frames. Here again, we experiment with a number of thresholds for garbage and non-garbage, namely: [0.95, 0.7], [0.9, 0.8], [0.9, 0.7], [0.9 0.6], [0.85, 0.65], [0.8, 0.7], [0.8, 0.6] and [0.7, 0.7].

1.3. AuToBI Features Experiments

In addition to using AuToBI for finding segments for rescoring, in this second experiment, we also use it to extract acoustic/prosodic features for classifying a frame directly. Knox et al. found that syllable like regions with approximately equal intensity and duration are correlated with laughter [5]. Based on these findings, we use AuToBI both to identify segments and to calculate average acoustic/prosodic values within each segment.

Our intention in including these features extracted over a full syllable region is to incorporate the broader prosodic acoustic context of the frame into the decision. To extend this beyond the current syllable, we also extract the same features from the preceding and following syllable.

The 9 acoustic features we extract are:

- Range Normalized Intensity and its Delta – the intensity contour (dB) is linearly normalized to a [0,1] range.
- Z-score Normalized Pitch (log Hz) and its Delta – the pitch contour is z-score normalized based on the mean and standard deviation of log Hz in the training data
- Mean Spectral Tilt and its Delta – spectral tilt is calculated as the average slope of the spectrogram within each 10ms frame contained in the syllable
- Duration (sec)
- Length of preceding and following pauses (sec)

Finally, we also observe that none of the development or training material begin with **laughter** or **filler**. To make the most use of this artifact, we also incorporate the position of the syllable in the stimulus.

The 9 acoustic features extracted over the current, previous and following syllable, plus the position of the current syllable leads to 28 features to describe the acoustic/prosodic content of the broader context of each frame.

We perform two experiments using these features. The first ("Prosody") uses the full feature set described in this section. The second ("Duration") uses only four features: the position in the stimulus, the duration of the current, previous and following syllables. This allows us to measure the impact of the segmentation in isolation of the other acoustic/prosodic information. In these experiments, we extend the baseline feature vector with its 141 features [3], with the above acoustic features. We then retrain a frame based SVM classifier using this augmented with the 28 additional features described here.

Results

The results of the rescoring procedure described in Section 3.2 are presented in Table 1.

We find that the rescoring procedure described in Section 3.2 consistently results in performance increases as measured by AUC and Uweighted Average AUC (UAAUC). We find that the improvement to performance is approximately equal whether using C=0.001 or 0.1, despite the fact that C=0.1 has a higher baseline performance. The improvement to **laughter** detection is slightly larger than the improvement to **filler** detection. This may be due to the fact that the unmodified laughter performance is lower than that of **filler**, or it may be insignificant noise.

Config	Complexity	Laughter	Filler	UAAUC
Baseline	0.001	84.4%	87.4%	85.90%
Rescoring	0.001	85.9% +1.5%	88.7% +1.3%	87.30% +1.4%
Baseline	0.1	86.2%	89.0%	87.60%
Rescoring	0.1	87.6% +1.4%	90.1% +1.1%	88.85% +1.25%

Table 2: *Impact of Syllable-based Rescoring with C=0.1 and 0.001. Results are reported on development data in AUC.*

We generally find that the thresholds for **laughter** and **filler** classes are best when set 0.2 points lower than for the **garbage** class and that the best performing combination of results at complexity 0.1 to be Average Segment Threshold = [0.9, 0.7]; Majority Class Threshold = [0.9, 0.7] and Label Exchange Threshold [0.9, 0.7].

These thresholding operations provide improvements in two ways. First, they encourage consistent labeling of frames drawn from the same syllabic region. Many of the errors that we observed in the first-pass predictions were short bursts of incorrect labels in otherwise correct areas. By identifying a syllabic region, and encouraging consistent labeling, we are able to "smooth out" some of these rough patches. Second, by using lower thresholds for the acoustic events, **laughter** and **filler**, compared with **garbage**, we increase the likelihood of hypothesizing these events. We require less evidence to make a prediction of an event of interest than we do for the majority, background class. Future work will involve the automated tuning of these threshold settings attempting to learn from the first-pass performance the optimal values to modify the confidence scores.

We next report the results of experiments incorporating longer range acoustic/prosodic features in the feature vector. Table 2 describes the performance using the full set of 28 features, and only the 4 duration based features on the development set. All results are based on SVM with C=0.1.

We find that the inclusion of the four duration features does not improve performance. To the contrary, we find that the duration features bring the performance back down to slightly below the baseline. However, adding the additional acoustic prosodic features generates a substantial improvement of 1.65% to the UAAUC. Based on the observation that the durational features seem to undo the gains of the rescoring mechanism, we evaluate the performance of the baseline feature set augmented with the 28 prosodic features. This results in performance that is marginally more effective (0.05% UAAUC) than the same features without rescoring. We hypothesize that this is because the acoustic/prosodic features already incorporate information about the

segmentation. By repeating features across frames drawn from the same syllabic segment, we have effectively encouraged the predictions within each syllable to be homogenous enough to render the rescoring approaches unnecessary.

Config	Laughter	Filler	UAAUC
Baseline	86.20%	89.00%	87.60%
Rescoring	87.60% +0.6%	90.10% +1.10%	88.85% +1.25%
Rescoring + Duration	86.10% -0.10%	89.0% +0.0%	87.55% -0.05%
Rescoring + Prosody	88.60% +2.4%	89.9% +0.9%	89.25% +1.65%
Prosody	88.70% +2.5%	89.9% +0.9%	89.30% +1.7%

Table 2: *Development Performance with Acoustic/Prosodic Features with absolute performance over baseline*

In Table 3, we report the performance of these approaches on the test set. In these experiments, we train models using only the train set, without inclusion of the development data.

Because we have a limited number of evaluations that could be performed on the test, we omit the rescoring experiments using only durational features.

Config	Laughter	Filler	UAAUC
Baseline	82.90%	83.60%	83.25%
Rescoring	83.80% +0.9%	84.22% +0.62%	84.01% +0.76%
Rescoring + Prosody	84.05% +1.15%	84.58% +0.98%	84.31% +1.06%
Prosody	84.64% +1.74%	85.06% +1.46%	84.85% +1.6%

Table 3: *Test Performance with Acoustic/Prosodic Features with absolute performance over baseline*

We find that the performance gains we observe by rescoring on the development set are similarly realized on the test set. When the prosodic and rescoring models are combined, we see a performance gain. Finally, we see performance gains in-keeping with our development gains on our Prosody-based model, our highest-performing model, which increased performance 1.6% over the baseline.

Conclusion and Future Work

We find that inclusion of information drawn from pseudosyllable segmentation can substantially improve the detection of laughter and filled pauses. We explored two approaches to make use of this information: 1) Rescoring frame level hypotheses based on the predictions elsewhere in the syllable, 2) inclusion of syllable-based acoustic/prosodic features to the frame-based feature vectors. We find that both of these approaches are able to improve detection UAAUC by between 1.25% and 1.70% on the development and 0.76% and 1.6% on the test segmentations of the SVC corpus.

In the future, we will extend this work to explore a more thorough set of acoustic/prosodic features. In this work, we have treated **laughter** and **filler** as single classes of acoustic events. However, many researchers have noticed that there is no stereotypical laughter that lends itself to acoustic modeling. We will explore modeling techniques that can take better advantage. In a similar vein, different languages express filled pauses differently, often with distinct segmental content. A valuable extension of this work will be to explore the applicability of these approaches to recognizing filled pauses in other languages.

Acknowledgements

This work was partially funded by DARPA FA8750-13-2-0041 under the Deep Exploration and Filtering of Text (DEFT) Program and Air Force Office of Scientific Research Grant #FA-9550-11-1-0120.

References

Esposito, A., et al., “Children Speech Pauses as Markers of Different Discourse Structures and Utterance Information Content”, (in coll. A. Esposito, G. Palombo), *Proceedings of the International Conference From Sound to Sense: + 50 years of discoveries in Speech Communication*, MIT, 2004.

Pon-Barry, H., et al., “Responding to Student Uncertainty in Spoken Dialogue Systems”, in *International Journal of Artificial Intelligence in Education 16*, pp. 171-194, 2006.

Schuller, B., et al., “The INTERSPEECH 2013 Computation Paralinguistics Challenge: Social Signals, Conflict, Emotion, Autism”, Proc. Interspeech 2013, ISCA, Lyon, France, 2013.

Knox, M., and Mirghafori, N., "Automatic laughter detection using neural networks," in *Interspeech*, 2007.

Kennedy, L. et al. (2004). “Laughter Detection in Meetings” Notebook Paper. *NIST ICASSP 2004 Meeting Recognition Workshop, Montreal*, May 2004.

Trouvain, J., “Segmenting Phonetic Units in Laughter”, in *Proc. 15th International Congress of Phonetic Sciences (ICPhS), Barcelona*, pp. 2793-2796, 2003.

Tanaka, H., and Campbell, N., “Acoustic Features of Four Types of Laughter in Natural Conversational Speech”, in *Proc. 17th International Congress of Phonetic Sciences*, pp. 1958-1961, 2011.

Schuller, B., et al., "Static and Dynamic Modelling for the Recognition of Non-Verbal Vocalisations in Conversational Speech", in *Perception in Multimodal Dialogue Systems, Proc. 4th IEEE Tutorial and Research Workshop on Perception and Interactive Technologies for Speech-based Systems (PIT 2008)*, pp. 99-110, 2008.

Goto M., et al., “A Real-time Filled Pause Detection System for Spontaneous Speech Recognition: Toward Spontaneous Speech Dialogue”, in *Proc. of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2000)*, pp.II-757-760, June 2000.

Villing, R., et al., “Automatic Blind Syllable Segmentation for Continuous Speech”, in *Irish Signals and Systems Conference (ISSC 2004)*, June 2004.

Rosenberg, A., “AuToBI – A tool for Automatic ToBI annotation”, in *Interspeech*, 2010.