COMPUTATIONALLY-EFFICIENT ENDPOINTING FEATURES FOR NATURAL SPOKEN INTERACTION WITH PERSONAL-ASSISTANT SYSTEMS

Harish Arsikere[1] *Elizabeth Shriberg*[2] *Umut Ozertem*[3]

[1]Electrical Engineering Department, University of California, Los Angeles, CA, USA
[2]Speech Technology and Research Laboratory, SRI International, Menlo Park, CA, USA
[3]Microsoft, Sunnyvale, CA, USA
harishan@g.ucla.edu, ees@speech.sri.com, umut.ozertem@microsoft.com

ABSTRACT

Current speech-input systems typically use a nonspeech threshold for end-of-utterance detection. While usually sufficient for short utterances, the approach can cut speakers off during pauses in more complex utterances. We elicit personal-assistant speech (reminders, calendar entries, messaging, search) using a recognizer with a dramatically increased endpoint threshold, and find frequent nonfinal pauses. A standard endpointer with a 500 ms threshold (latency) results in a 36% cutoff rate for this corpus. Based on the new data, we develop low-cost acoustic features to discriminate nonfinal from final pauses. Features capture periodicity, speaking rate, spectral constancy, duration/intensity, and pitch of prepausal speech – using no speech recognition, speaker or session information. Classification experiments yield 20% EER at a 100 ms latency, thereby reducing both cutoffs and latency compared with the threshold-only baseline. Additional results on computational cost, feature importance, and speaker differences are discussed.

Index Terms— endpointing, acoustic-prosodic features, personal assistants, pausing, computationally efficient

1. INTRODUCTION

End-of-utterance detection (or speech endpointing) is an important initial processing step in human-computer dialog systems. An ideal endpointing mechanism (a) does not miss any speech from the user, and (b) stops listening shortly after the utterance has ended. Most systems employ a simple pause-length threshold (typically 500 ms to 1 second) for endpointing [1]. While the approach works reasonably well for short utterances, it can cut speakers off during pauses in more complex utterances – slowing an interaction and frustrating users [2]. The frequency and duration of pausing is expected to be particularly high in cases where speakers are distracted by other real-world tasks, particularly for mobile applications. Other factors that increase internal pausing are waiting for dynamic screen content, deciding among many alternatives, adding multiple facets to a query, and leaving more complex messages in voice dictation. Simply increasing the pause threshold is not a viable solution, since this increases system latency at true utterance ends.

Previous studies used lexical and acoustic-prosodic cues for speech endpointing [3, 4] in an early human-computer dialog system. Related work has used similar features for disfluency detection, emotion classification, detection of developmental disorders, speaker-turn identification and speech/nonspeech classification, in both human-computer and human-human dialog [5–18]. Lexical cues include word ngrams; prosodic cues include silence duration, vowel or syllable duration, pitch (F_0) and formant frequencies. A few studies (e.g., [15]) have also used large-scale acoustic feature sets extracted with the openSMILE toolkit [19].

The present study aims to develop features for improved endpointing in present-day personal-assistant technology. Our goal is to design an endpointing mechanism that triggers whenever a nonspeech segment of p ms (where p is as small as possible) is detected, and estimates, based on the *pre-pausal* speech, whether or not the speaker is done. Our approach is novel with respect to prior work in a number of ways. We are particularly interested in the potential of nonlexical features, since endpointing is typically done on a user's device before speech is sent to a remote server for recognition. Nonlexical features also offer potential domain independence, which is important for broad coverage over different application domains. Because of the task definition, all features must be purely causal, or based only on speech occurring before the pause in question. For practical reasons, we consider only features that do not rely on speaker or session normalization, since this information complicates an implementation and may not always be available. Features may differ by speaker (and we indeed find this is the case), but their computation does not rely on this information.

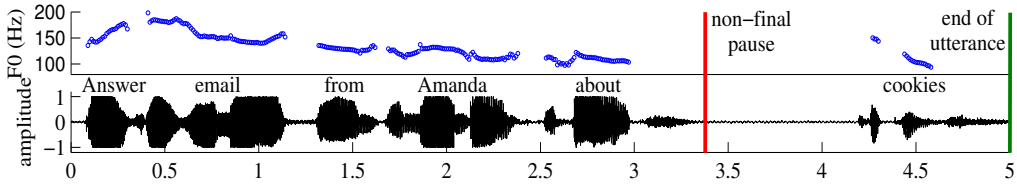
Finally, unlike past work, we collect a new database of personal-assistant speech (via prompt-based elicitation), and do so with a modified endpoint threshold of 5 seconds. This is done for two (related) reasons. First, current commercial systems typically use endpoint thresholds of 1 second or less (larger thresholds produce too much latency for true utterance ends). Data collected with such systems are not useful for our purpose, because several utterances would only be partially saved owing to speaker cutoffs. Moreover, there is no reliable way of judging whether or not a given recording is incomplete. Second, speakers quickly adapt to systems that cut them off, by waiting longer to start, or breaking down their utterances into smaller chunks. In other words, they do not behave the same way as they would if they did not fear being cut off.

2. METHOD

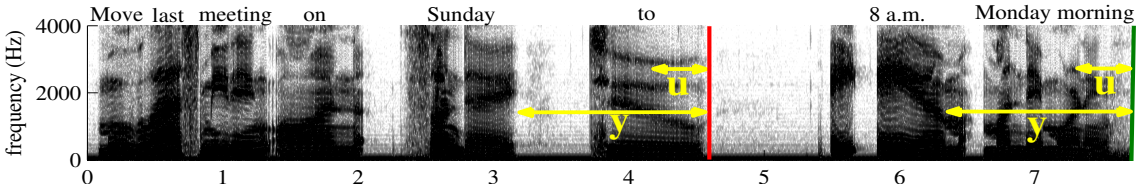
2.1. Data

We developed an elicitation method to collect complex utterances. A tool reads prompts from a text file and displays them in random order over subjects. The subjects are all adult, native speakers of American English. They are instructed to speak as if they were talking to their personal assistant, and to try to sound as natural as possible. Once a prompt is displayed, the subjects are given 3 seconds to start speak-

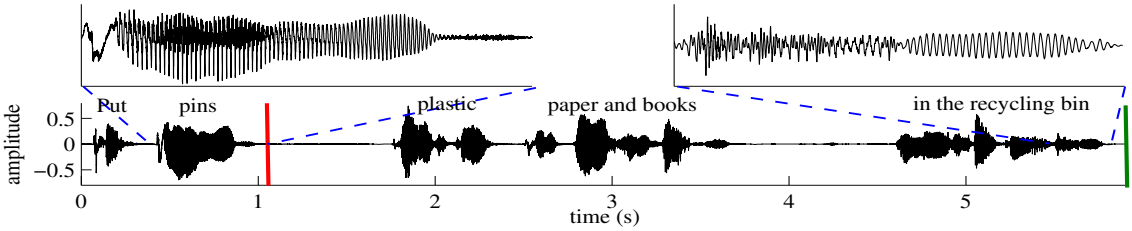
This research was conducted while the first author was an intern at Microsoft and the second author was with Microsoft Research (Silicon Valley).



(a) Nonfinal pauses are typically preceded by steadier pitch contours, longer segments with continuous voicing, and syllables with higher intensity, when compared to utterance ends.



(b) Looking at short time intervals (denoted by **u**), nonfinal pauses are usually preceded by more constant vocal-tract features when compared to utterance ends. Sometimes, the two classes also differ with regard to spectral modulation over long time intervals (denoted by **y**).



(c) Syllables preceding nonfinal pauses are usually more periodic compared to syllables before utterance ends.

Fig. 1: Illustration of selected features.

ing, using any content from their personal lives that would make sense. All prompts begin with a few *keywords*, encouraging subjects to start speaking without planning their utterances completely; for example: "*Answer email from <PERSON> about <TOPIC>*". Responses are recorded in a quiet environment using a close-talking microphone, at a sampling rate of 16 kHz and a resolution of 16 bits/sample. Recognition output is also displayed (using a message-dictation back-end) to simulate the feedback mechanism of a dialog system. This is important: users saw the same output as with an actual system, including errors, but the system waited ten times as long before cutting off during a pause. Note that the recognition hypothesis is presented only after the subject has stopped speaking, and that subjects are not allowed to respond to the recognition errors they observe. The database contains a total of 5297 utterances from 34 speakers; the number of utterances per speaker varies from 82 to 180. We collected many utterances per speaker for two reasons. First, we wanted speakers to learn that the system did not cut them off as quickly as typical speech dialog systems. Second, we were interested in whether speakers differ in pause behaviors.

We define a *nonfinal pause* as a mid-utterance silence that is at least 100 ms long – this is about the shortest duration we can choose; shorter pauses may correspond to stop bursts. Roughly 70% of the utterances in our corpus have at least one nonfinal pause, and the pause duration ranges from 100 ms to 10.7 seconds (median value = 380 ms). For convenience we use word alignments from a message-dictation back-end to locate nonfinal pauses; obviously in practice a nonspeech detector would be used. While this discrepancy could somewhat affect results, we assume that advanced endpointing features would only be used in conjunction with a good nonspeech detector, and that results for pause detection would be roughly comparable for cleaner signals. For noisier signals, all features would be affected, a topic beyond the scope of the current paper.

2.2. Features

Based on prior work on pausing, as well as on inspection of our new data, we study features grouped into five types: (1) *pitch trends*, (2) *duration and intensity*, (3) *spectral constancy*, (4) *speaking rate*, and (5) *periodicity*. A brief motivation for each type is given below.

• ***Pitch trends***: F_0 remains fairly steady before nonfinal pauses, but typically falls and/or fluctuates more at utterance ends (see Fig. 1a). This feature type models intonation patterns, with a normalization to account for the speaker's 'baseline' F_0.

• ***Duration and intensity***: Syllable intensity tends to drop more at utterance ends (because of reduced vocal effort), and continuously-voiced segments tend be longer before nonfinal pauses (see Fig. 1a). This feature type models these two phenomena, with a normalization to account for the speaker's 'baseline' intensity.

• ***Spectral constancy***: Speakers appear to maintain a more fixed vocal-tract configuration before nonfinal pauses, especially in syllable-final phonemes (both voiced and unvoiced). This phenomenon can be discerned from spectrograms (see Fig. 1b); we model it by analyzing the signal over short time intervals.

• ***Speaking rate***: Some speakers tend to lower their speaking rate as they approach a nonfinal pause, presumably to gain time to plan content. We model speaking rate using the amplitude modulation of spectral components over long time intervals (see Fig. 1b).

• ***Periodicity***: We observed in our corpus that voiced segments before nonfinal pauses are, in general, more regular and periodic compared to voiced segments at utterance ends (Fig. 1c). We attribute the aperiodicity in utterance-final syllables to a possible reduction in subglottal pressure, which makes it difficult to sustain regular vocal-fold oscillations.

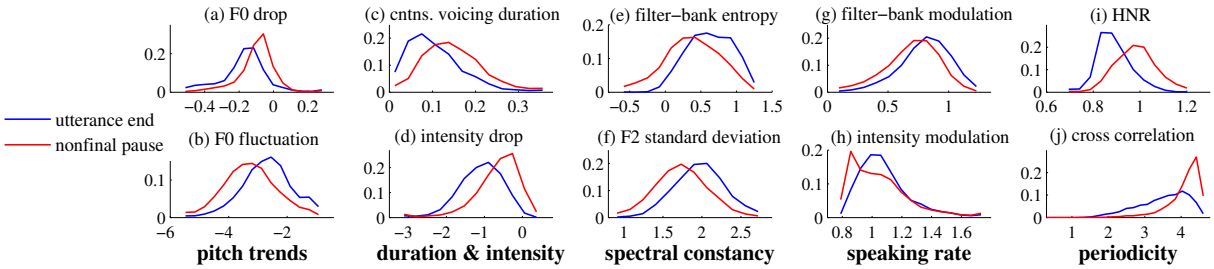


Fig. 2: Distributions for the five feature types (2 representative features per type), obtained by pooling results over all speakers.

Our system uses 15 features in total. Due to space limitations, we discuss the step-wise implementation of two representative features from each of the feature types. Note that all features are computed using only the given utterance (from the beginning up to the 100 ms-long silence). Since the phenomena just discussed tend to be distributed over time and frequency, the algorithms are designed to capture the combined effect of several short-term observations via a statistic such as the *median* or the *minimum*. The dynamic ranges of all features are reduced using $\log()$ or power-law compression.

• ***Pitch trends – F_0 drop***: (1) Using the Snack Sound Toolkit (*get_f0*) [20], obtain F_0 and the corresponding voicing decision at 10 ms intervals. (2) Divide the utterance into segments that are continuously voiced. (3) Compute the log ratio of the minimum F_0 in the last segment to the median F_0 over all previous segments.

• ***Pitch trends – F_0 fluctuation***: (1) Obtain the F_0 contour and voiced segments using Snack. (2) Compute the Hadamard transform [21] of the last 16 F_0 values in the last voiced segment. (3) Find the log ratio of $(h_1^2 + h_2^2)$ to $\sum_{k=0}^{15} h_k^2$, where $\{h_0, h_1, \ldots, h_{15}\}$ denote the transform coefficients (note that the k^{th} Hadamard basis function has k zero crossings).

• ***Duration and intensity – continuous voicing duration***: (1) Obtain voiced segments using Snack. (2) Compute $\log((M-1) \times 0.01)$, where M is the number of frames in the last voiced segment (with an inter-frame spacing of 10 ms).

• ***Duration and intensity – intensity drop***: (1) Compute an intensity (energy) contour using 20 ms frames at 10 ms intervals. (2) Smooth the intensity contour using a 5-point moving-average filter. (3) Detect the peaks in the smoothed contour, and discard those that are within 100 ms of a higher peak; the remaining ones correspond roughly to syllable locations. (4) Compute the log ratio of the final syllable peak to the median of all previous syllable peaks.

• ***Spectral constancy – filter-bank entropy***: (1) Consider the last 500 ms of the signal. Divide it into 200 ms chunks with 100 ms overlap. (2) For the i^{th} chunk (i varying from 1 to M): divide into 20 ms frames with 10 ms overlap; compute the FFT magnitude spectrum of each frame and pass it through a 26-channel Mel filterbank; find the time variance of each channel output; compute $\Sigma^{(i)}$, the average variance across channels. (3) Find the log minimum of $\{\Sigma^{(1)}, \Sigma^{(2)}, \ldots, \Sigma^{(M)}\}$.

• ***Spectral constancy – F_2 standard deviation***: (1) Obtain voiced segments using Snack. Divide the last voiced segment into 200 ms chunks with 100 ms overlap. (2) For the i^{th} chunk (i varying from 1 to M): use Snack to estimate the second formant frequency (F_2) at 10 ms intervals; compute $\sigma^{(i)}$, the standard deviation of F_2 over time. (3) Find the log minimum of $\{\sigma^{(1)}, \sigma^{(2)}, \ldots, \sigma^{(M)}\}$. Similar features can be computed using F_1 and F_3.

• ***Speaking rate – filter-bank modulation***: (1) Divide the last 1 second of the signal into 20 ms frames with 10 ms overlap ('sampling rate' = 100 Hz). (2) Compute the FFT magnitude spectrum of each frame and pass it through a 26-channel Mel filter-bank. (3) For the i^{th} channel (i varying from 1 to 26): compute the FFT magnitude spectrum of channel output (this is essentially the amplitude modulation spectrum with a 50 Hz bandwidth); compute $\rho^{(i)}$, the percentage energy above 10 Hz in the modulation spectrum. (4) Find the log average of $\{\rho^{(1)}, \rho^{(2)}, \ldots, \rho^{(26)}\}$.

• ***Speaking rate – intensity modulation***: (1) Divide the last 1 second of the signal into 300 ms chunks with 100 ms overlap. (2) For the i^{th} chunk (i varying from 1 to M): obtain a smoothed intensity contour (as described earlier); compute $\nu^{(i)}$, the percentage energy above 4 Hz in the modulation spectrum of the intensity contour. (3) Find the log maximum of $\{\nu^{(1)}, \nu^{(2)}, \ldots, \nu^{(M)}\}$.

• ***Periodicity – HNR***: (1) Obtain voiced segments using Snack. (2) Divide the last voiced segment into 60 ms frames with 10 ms overlap. (3) Estimate the harmonic-to-noise ratio (HNR) of each frame in the last segment using the algorithm proposed in [22]. (4) Compute the 75$^{\text{th}}$ percentile of the frame-wise HNRs.

• ***Periodicity – cross correlation***: (1) Using Snack, obtain voicing decision and the corresponding value of normalized cross correlation [23] at 10 ms intervals. (2) Compute the cube root of the percentage number of frames in the last voiced segment with a normalized cross correlation greater than 0.9.

Figure 2 shows the distributions obtained by pooling results from all 34 speakers. Except for *speaking rate*, all feature types provide reasonably good separation between nonfinal pauses and utterance ends; this will be discussed further in Sec. 3.4.

2.3. Classification experiments

Our task is to determine whether the speaker is done (negative class, since our target class is nonfinal pauses) or not done (positive class), whenever a silence of 100 ms is encountered. In total, our database has 5297 instances of utterance ends and 8061 instances of nonfinal pauses. Experiments use leave-one-out cross validation – leaving out data from one speaker in the training phase (in order to be used for evaluation), and iterating the process over all the speakers in the database. Note that data from the same speaker are never used for both training and evaluation. Support vector machines (SVMs), as implemented in LIBSVM [24], are used as classifiers. All 15 features are scaled to lie in the range [-1,1], prior to training and evaluation. The standard radial-basis-function kernel is used, and the optimal values of C (the penalty paramter) and γ (the kernel parameter) are determined via a two-dimensional grid search: C is chosen from $\{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$ and γ is chosen from $\{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$.

3. RESULTS AND DISCUSSION

3.1. Results overall and by speaker

Figure 3a shows receiver operating characteristic (ROC) curves – plots of correct detection probability (P_d) versus false alarm probability (P_f) – based on overall classification results and on results

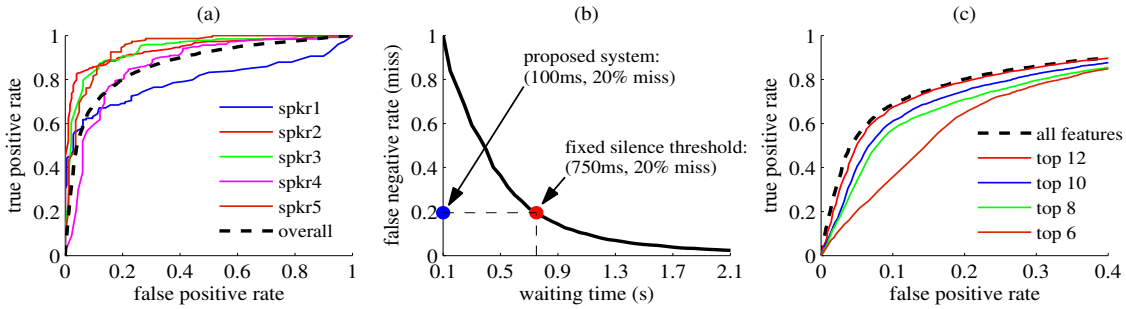


Fig. 3: Results for binary classification. "True positive" = system correctly detects nonfinal pause, waits for user to continue. "False negative (miss)" = system misses nonfinal pause and incorrectly endpoints, prematurely cutting off user. (a) ROC curves illustrating speaker variability. (b) Comparing the fixed-silence-threshold mechanism with the proposed approach. (c) ROC curves for the top-N (N least-complex) features.

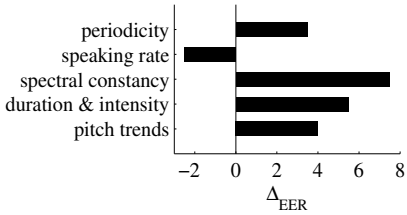


Fig. 4: Relative importance of the feature types.

for six speakers in our database. The overall equal error rate (EER) is 19.9%. It is clear from the ROCs that there are differences by speaker, warranting further study on speaker adaptation. Informal listening suggests true speaker differences in how nonfinal pauses are rendered acoustically. For example some speakers marked nonfinal pauses by preceding pitch rises whereas others did not.

3.2. Comparison with fixed-silence-threshold baseline

To compare performance to a baseline system that uses a fixed pause threshold, we calculate the false negative rate (nonfinal pauses misclassified as final, or "cutoffs") as a function of the pause-length threshold (or the *waiting time*); the resulting curve is shown in Fig. 3b. As indicated by the red circle, the baseline mechanism has to wait for 750 ms (before endpointing) in order to match the EER performance of the proposed system, whose waiting time is just 100 ms (indicated by the blue circle). In other words, our approach can match the baseline system in performance while achieving much lower latency (about one-eighth in this case). An additional comparison is at 500 ms, a common silence threshold for deployed systems. A standard system using a 500 ms latency results in premature cutoffs for 36% of the utterances in our corpus; in comparison the proposed system cuts off only 20% at a latency of only 100 ms.

3.3. Analysis by feature cost

As noted earlier, our end goal is practical online implementation, so feature cost is an important consideration. We looked at the performance-cost trade-off by simply ranking features according to our estimates of computational complexity, and running classification experiments with the top-N features. Simple signal processing steps such as Mel filter-bank processing and intensity contour estimation are 'cheap'; voiced/unvoiced classification is 'moderately expensive'; F_0 and HNR estimation are 'expensive'; and formant estimation is 'very expensive'. If two features share the same kinds of operations, the more complex feature is that which requires normalization with respect to a baseline and/or a large buffer duration.

Fig. 3c shows ROC curves for the top-N features (N = 15, 12, 10, 8, 6). Discarding the three most expensive features (which happen to be *formant standard deviations*) has little effect on performance; the *filter-bank entropy* feature appears to be adequate for modeling *spectral constancy*. The top-10, top-8 and top-6 features yield EERs of 22.7%, 24.8% and 26.4%, respectively. For resource-constrained applications, the top-10 features seem to offer a good trade-off point.

3.4. Relative importance of feature types

An assessment of the relative contributions of the proposed feature types can guide us towards improved feature modeling schemes (feature weighting, for example). To quantify the importance of a feature type, we measure the relative increase in EER (denoted by Δ_{EER}) that is incurred by removing it from the complete feature set; see Figure 4. While *spectral constancy* is the most useful feature type, *duration and intensity*, *pitch trends* and *periodicity* also make substantial contributions. However, *speaking rate* is not useful – inconsistent with studies of human-human hesitation. A possible explanation is that speech timing is less helpful in human-computer dialog, because speakers use a slower and more deliberate style overall when talking to a system that makes recognition errors.

4. CONCLUSION

Through our newly-collected database of personal-assistant speech, we find that speakers pause quite frequently during their utterances, especially while composing long, complex messages that are not planned *a priori*. In order to minimize cutoffs (due to long pauses) without incurring high latency, we need an approach that is causal, online and computationally efficient. We explored five types of acoustic-prosodic features based on pitch, intensity, voicing, short-term and long-term spectral characteristics, and glottal phenomena. The features do not need information from speech recognition, and do not rely on speaker or session statistics. Compared to a standard endpointer, which yields a 36% cutoff rate at 500 ms latency, our approach reduces cutoffs to less than 20% at only 100 ms latency. Speaker-dependent results suggest that further benefit might be obtained using speaker-adaptive modeling, for relevant applications. Finally, an analysis of performance by feature cost reveals that good performance can be achieved using fewer, cheaper features. Future work involves experimenting with noisy utterances and with prompts that are designed to elicit more free-form speech.

5. ACKNOWLEDGMENT

We sincerely thank Karthik Raghunathan for his work on the data-collection tool, Kjel Larsen and Aidan Brennan for their data-collection efforts, and Abhik Lahiri and Ashley Fidler for their help and suggestions during the project.

6. REFERENCES

[1] R. Hariharan, J. Häkkinen, and K. Laurila, "Robust end-of-utterance detection for real-time speech recognition applications," in *Proceedings of ICASSP*, 2001, pp. 249–252.

[2] Jiepu Jiang, Wei Jeng, and Daqing He, "How do users respond to voice input errors? Lexical and phonetic query reformulation in voice search," in *Proceedings of SIGIR*, 2013.

[3] L. Ferrer, E. Shriberg, and A. Stolcke, "Is the speaker done yet? Faster and more accurate end-of-utterance detection using prosody in human-computer dialog," in *Proceedings of ICSLP*, 2002, pp. 2061–2064.

[4] L. Ferrer, E. Shriberg, and A. Stolcke, "A prosody-based approach to end-of-utterance detection that does not require speech recognition," in *Proceedings of ICASSP*, 2003, pp. 605–608.

[5] D. O'Shaughnessy, "Recognition of hesitations in spontaneous speech," in *Proceedings of ICASSP*, 1992, pp. 521–524.

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

[7] 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 Transactions on Audio, Speech, and Language Processing*, vol. 14, pp. 1526–1540, 2006.

[8] Sankaranarayanan Ananthakrishnan, Prasanta Ghosh, and Shrikanth Narayanan, "Automatic classification of question turns in spontaneous speech using lexical and prosodic evidence," in *Proceedings of ICASSP*, 2008, pp. 5005–5008.

[9] K. Audhkhasi, K. Kandhway, O. D. Deshmukh, and A. Verma, "Formant-based technique for automatic filled-pause detection in spontaneous spoken English," in *Proceedings of ICASSP*, 2009, pp. 4857–4860.

[10] A. Hjalmarsson and K. Laskowski, "Measuring final lengthening for speaker-change prediction," in *Proceedings of Interspeech*, 2011, pp. 2065–2068.

[11] W. Wang, K. Precoda, C. Richey, and G. Raymond, "Identifying agreement/disagreement in conversational speech: a cross-lingual study," in *Proceedings of Interspeech*, 2011, pp. 3093–3096.

[12] C. T. Ishi, H. Ishiguro, and N. Hagita, "Analysis of acoustic-prosodic features related to paralinguistic information carried by interjections in dialogue speech," in *Proceedings of Interspeech*, 2011, pp. 3133–3136.

[13] H. Medeiros, H. Moniz, F. Batista, I. Trancoso, and L. Nunes, "Disfluency detection based on prosodic features for university lectures," in *Proceedings of Interspeech*, 2013, pp. 2629–2633.

[14] Andreas Tsiartas Maarten Van Segbroeck and Shrikanth S. Narayanan, "A robust frontend for vad: exploiting contextual, discriminative and spectral cues of human voice," in *Proceedings of Interspeech*, 2013, pp. 704–708.

[15] Rahul Gupta, Kartik Audhkhasi, Sungbok Lee, and Shrikanth Narayanan, "Paralinguistic event detection from speech using probabilistic time-series smoothing and masking," in *Proceedings of Interspeech*, 2013, pp. 173–177.

[16] Meysam Asgari, Alireza Bayestehtashk, and Izhak Shafran, "Robust and accurate features for detecting and diagnosing autism spectrum disorders," in *Proceedings of Interspeech*, 2013, pp. 191–194.

[17] Gábor Gosztolya, Róbert Busa-Fekete, and László Tóth, "Detecting autism, emotions and social signals using AdaBoost," in *Proceedings of Interspeech*, 2013, pp. 220–224.

[18] Okko Räsänen and Jouni Pohjalainen, "Random subset feature selection in automatic recognition of developmental disorders, affective states, and level of conflict from speech," in *Proceedings of Interspeech*, 2013, pp. 210–214.

[19] Florian Eyben, Martin Wöllmer, and Björn Schuller, "openSMILE: the Munich versatile and fast open-source audio feature extractor," in *Proceedings of the International Conference on Multimedia*, 2010, pp. 1459–1462.

[20] K. Sjölander, "The Snack sound toolkit," *KTH, Stockholm, Sweden (Online: http://www.speech.kth.se/snack/)*, 1997.

[21] A. K. Jain, *Fundamentals of digital image processing*, Prentice-Hall Englewood Cliffs, 1989.

[22] Y. Qi and R. E. Hillman, "Temporal and spectral estimations of harmonics-to-noise ratio in human voice signals," *The Journal of the Acoustical Society of America*, vol. 102, pp. 537–543, 1997.

[23] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," *Speech coding and synthesis*, 1995.

[24] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at: http://www.csie.ntu.edu.tw/∼cjlin/libsvm.