ROBUST SPEECH DETECTION AND SEGMENTATION FOR REAL-TIME ASR APPLICATIONS

Izhak Shafran & Richard Rose

AT&T Labs Research, Florham Park, NJ 08873

ABSTRACT

This paper provides a solution for robust speech detection that can be applied across a variety of tasks. The solution is based on an algorithm that performs non-parametric estimation of the background noise spectrum using minimum statistics of the smoothed short-time Fourier transform (STFT). It will be shown that the new algorithm can operate effectively under varying signal-to-noise ratios. Results are reported on two tasks – HMIHY[1] and SPINE[2], which differ in their speaking style, background noise type and bandwidth. With a computational cost of less than 2% real-time on a 1GHz P-3 machine and a latency of 400ms, it is suitable for real-time ASR applications.

1. INTRODUCTION

The goal of the work described in this paper is to develop a speech detection method that can be applied over a range of conditions and applications without adjustments. The difficulty of the task is compounded by the fact that the objectives for speech detection are different for different applications.

In applications involving human-human interaction, the input may contain multiple sentences. In such cases, the system designer must weigh the effect of sending intervals of environment noise to the decoder against the effects of breaking contiguous utterances into multiple phrases. The SPINE corpus is an example of such a domain where robust segmentation algorithms have been shown to improve ASR performance, particularly in certain difficult noise conditions [2, 3].

On the other hand, in applications where a machine prompts a human, the input speech is naturally segmented into specific responses from the user. By removing long non-speech intervals from the ASR input, a good segmentation algorithm can save considerable load on a server that is processing multiple clients. In addition, a general-purpose segmentation algorithm should be capable of operating in conditions where the signal-to-noise ratio varies considerably and the input is corrupted by noise such as telephone tones and background hum.

The rest of the paper is organized as follows. Section 2 provides a brief background and a few examples of current techniques. It also includes a short discussion on the choice of evaluation criterion. Although the paper aims at providing speech segments to the ASR, to assess the difficulty of the task, frame-level classifications were performed initially. These exploratory experiments are briefly described in section 3. Section 4 delves into the solution based on extreme statistics, and explains the motivation and details of the approach. Experiments and results on two different tasks are reported in section 5 which is followed by conclusions in section 6.

2. BACKGROUND

2.1. Current Approaches

In the last few years, a variety of segmentation algorithms have been proposed and shown to work on specific tasks. Yet, a robust and general-purpose solution for real-time ASR is lacking.

Typically, a segmentation scheme for real-time application consists of three parts, as shown in figure 1. Frames are extracted (typically 10ms long), then a core module decides whether a particular frame is speech or non-speech. The sequence of frame-level decisions are converted into utterance or segment boundaries using a simple state machine which pads the boundaries to account for low-energy components of speech such as fricatives, voice onsets or short pause between words.

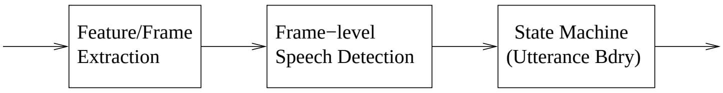

Fig. 1. An illustration of a typical segmentation algorithm.

A number of current segmentation algorithms for real-time ASR are based on techniques developed in the speech coding community where voice activity detection (VADs) are popularly used for identifying non-speech segments that need not be transmitted (e.g. [4]). The VADs were originally designed to meet strict latency requirements and constraints of low computational power. Typically, in a VAD-based segmentation algorithm, the first few frames are assumed to be noise and a threshold is computed from it. Any frame with more energy than the threshold is marked as speech. The threshold is continually updated using the energy level of the non-speech frames. Segmentation of the input waveform has also been treated as an edge-detection problem (e.g. [5]). Instead of relying on a threshold, the onset of speech is detected by a matched filter with a pre-determined slope which locates sudden change in signal energy. These methods are not inherently robust to noise bursts, telephone tones, and varying noise powers, and require additional heuristics in real applications.

In ASR, model-based segmentation such as [3] and [2], use hidden Markov model (HMMs) to model the dynamic nature of speech. The input feature could be standard cepstral vectors (e.g. [2]), or special features such as normalized cross correlations (e.g. [3]). The segmentations are produced by a Viterbi algorithm over the complete sequence of input vectors. This makes them unsuitable for real time ASR applications. In certain real-time ASR tasks such as close captioning of broadcast news, it is possible to avoid explicit segmentation at the front-end. Instead, the complete input sequence is sent to the decoder, and the decoder uses partial

[1] "How May I Help You", a task that routes telephone calls from customers in an AT&T customer service application [1].

[2] Speech in Noisy Environments, a task designed by Naval Research Laboratory for research in robust ASR.

back-trace of continuous recognition to put out recognized word sequences or segments [6, 7]. When the input contains background noise that are not seen in the training data, this method could potentially increase ASR errors. Moreover, the need to operate a decoder continuously on each input makes this approach infeasible for a number of applications.

2.2. Evaluation Criterion

Unfortunately, there is no standard criterion to evaluate a segmentation algorithm. The difficulty arises largely from the lack of a clear definition for a segment. It is not clear whether linguistic criteria should be used in defining segment boundaries or whether elimination of long inter-word silence should be the sole criterion. In either case, the notion of a "true" segmentation is ill-defined, since humans themselves are inconsistent in identifying certain word boundaries. Taking a practical approach for this work, segments are assumed to be a run of words surrounded by non-speech intervals approaching several hundred milliseconds in duration.

Even if a reference were provided, there is no standard metric to compare segmentation algorithms. In previous work, performance has been reported in terms of a number of criteria – frame-level classification error, frame detection and false alarm rates, mean squared error in locating the edges of segments, and ASR word error rate. Usually, a combination of these measures are used to compare segmentations. Since this paper addresses segmentation for ASR applications, word error rate (WER) measured on the resulting segments will be used as a metric for all experiments in section 5. Additionally, since segmentation algorithm aims to reduce computational load by removing non-speech intervals from the input stream, the total number of frames passed to the ASR decoder will also be used as a metric.

3. FRAME-LEVEL CLASSIFICATION

To investigate the problem of classifying speech at a frame-level, a few exploratory experiments were conducted. A set of popular machine learning methods were applied. All of these methods assume the input features to be independent identically distributed processes. Although these methods have a few deficiencies, which are mentioned later in the section, they provide complex non-linear classification boundaries for separating speech from non-speech.

To encompass a variety of noise conditions, a 2.6 hour subset of the SPINE corpus was used for the task. For creating a reference, word-level segmentations were generated automatically using a state-of-the-art ASR system and then hand-corrected at word-boundaries. After excluding three transitional frames before and after a word boundary, the data was divided into 322K frames for training, 278K frames for validation, and 270K frames for testing.

A set of classifiers were trained using the training and the validation sets. The classifiers included – (a) Bayes classifier with Gaussian Mixture Models (GMM), (b) Multi-layer perception (MLP), (c) Bagging with MLPs, (d) Boosting with MLPs, (e) Mixture of Experts with MLPs, and (f) SVM [8]. These classifiers were trained and tested using "Torch", a software that is widely used in machine learning community.

Mel-warped cepstral coefficients (MFCC) were used as feature vectors, since they are popular for ASR applications. The raw MFCC features were used in the form available at the ASR front-end, i.e., without applying any batch processing such as cepstral variance normalization and vocal tract length normalization. Cepstral mean subtraction was also not performed.

The classification results are shown in the table 1. The test set contained 26% speech, and "–" in the table denotes performance below chance. The parameters of the classifiers were tuned using the validation set, and the optimal settings are reported under the corresponding frame error rate. The number of hidden units are denoted by "nhu", the number of mixtures in GMM by "ng", the number of MLP classifiers in bagging by "nb". The second column in the table shows the results of using static cepstral features. The third column reports the results when input cepstral features were appended with their first and second order difference. The best results of 4.2% frame error rate was obtained using delta cepstrum by bagging with MLPs. Adaboost did not perform above chance and the optimization routine in SVM could not handle the amount of data in the test set. Appending larger number of neighboring frames to the features, or applying ASR-trained linear discriminant analysis did not yield any further improvements in the classification performance.

Classifier	Features w/o deltas	Features w/ deltas
Bayes with GMM	14.9% (ng=16)	7.3% (ng=20)
MLP	7.3% (nhu=4)	–
Bagging+MLP	6.4% (nhu=6,nb=14)	4.2% (nhu=6,nb=16)
MixOfExperts+MLP	15.5%	–

Table 1. Frame error rate for classifying frames into speech and non-speech.

These static classifiers do not outperform algorithms based on signal processing techniques in [9] (4.9% FER) by a margin wide enough to warrant their use despite their deficiencies. One of the significant disadvantages that restricts their portability is the need for training samples of a variety of noise types. Moreover, they are inherently not capable of utilizing the temporal properties of the input sequence.

4. SEGMENTATION WITH MINIMUM STATISTICS IN STFT DOMAIN

4.1. Motivation

Segmentation methods such as the VAD-based methods described in Section 2.1 use heuristics to estimate the noise floor to decide the presence of speech. In place of applying heuristics, we take advantage of more rigorous techniques developed in the speech enhancement community where noise estimation has been studied for a number of years. However, the application of background noise estimation to ASR differs from that of speech enhancement in a number of ways. Most significantly, unlike speech enhancement, the reconstructed waveform is not required in ASR and latency requirements are on the order of hundreds of milliseconds.

Among the various techniques that are currently available, we chose a non-parametric estimation of noise spectrum which uses minimum statistics. The motivation for this technique comes from the following considerations. Speech and noise can be assumed to be statistically independent. Focusing only on additive noise, both speech and noise are positive and additive in power spectrum domain. Due to the highly non-stationary nature of speech, the speech energy in a given frequency bin is likely to fall to zero at sometime over an interval of a second or more. When the speech energy falls to zero, the signal energy is solely from the background noise. Hence, by tracking the minimas in each spectral bin over a sufficiently long interval, one can obtain statistics of the noise spectrum. From these statistics, the mean noise spectrum can be computed. To deal with non-stationary noise, the estimate can be updated continually after each input frame. In practice, the time-varying power spectrum of the input signal can be approximated by a smoothed short-time Fourier transform (STFT).

4.2. Estimation of Noise Spectrum

The utterance segmentation algorithm has three major components. These include estimation of the background noise spectrum, classification of frames into speech or non-speech, and determination of utterance boundary. This section focuses mainly on the background noise estimation algorithm which is performed in three steps. First, a smoothed estimate of the STFT magnitude is computed. Second, the spectral minimum statistics are tracked over a block of STFT frames corresponding to approximately a second of speech. Third, the average background noise statistics are estimated from the spectral minima using order statistics.

For each 10ms frame, the magnitude STFT, $Y(k,t)$ is computed where the indices k and t denote frequency and time respectively. The variations in the STFT are smoothed using a leaky integrator to obtain $Y_s(k,t)$. The coefficients of the integrator $\alpha(k,t)$ are continually updated for each frequency with the aim of reducing the average error between smoothed STFT and the raw STFT. This allows the smoothed STFT to quickly rise to the level of input speech signal.

$$Y_s(k,t) = \alpha(k,t)Y_s(k,t-1) + (1-\alpha(k,t))Y(k,t) \qquad (1)$$

The spectral minimum $M(k,t)$ is computed over a set of previous D smoothed STFT frames.

$$M(k,t) = \min_{\tilde{t}=t-D:t} Y_s(k,\tilde{t}) \qquad (2)$$

This spectral minimum is an example of extreme statistics, and it is lower than the mean of the noise, $\hat{N}(k,t)$. Fortunately, this can be corrected by applying a bias factor, $\hat{N}(k,t) = B_c(k,t)M(k,t)$, where the bias $B_c(k,t)$ can be computed using asymptotic approximations [10, 11]. As it turns out, the bias for each spectral bin can be approximated as function of its “equivalent degrees of freedom”, $\tilde{Q}(k,t)$ and D [12], as $B_c(k,t) \approx 1 + 2(D-1)/\tilde{Q}(k,t)$. The quantity, $\tilde{Q}(k,t)$, depends on the variance of current estimate of the noise $\hat{N}(k,t)$ and the smoothed STFT, $\tilde{Q}(k,t) = 2\text{var}^2\{\hat{N}(k,t)\}/\text{var}\{Y_s(k,t)\}$. The variance of the smoothed STFT is computed using a leaky integrator and updated after each input frame is received.

The original algorithm applies the bias equally to the speech and non-speech signal. Since the bias is computed as a function of the input variance over an interval, it is likely to have a higher value when speech is present. This causes an overshoot of the bias on steep spectral minimas which are buried in speech. This can be reduced significantly by modifying the bias to $B_{cs}(k,t)$ using a sigmoid function of short-term “a posteriori” SNR, R, as given below.

$$B_{cs}(k,t) = B_c(k,t)(1+\exp^{3(R-3)})^{-1} + (1+\exp^{3(3-R)})^{-1} \qquad (3)$$

This function does not effect low SNR noise, but clamps the bias to unity when speech is present, thus improves the ability of the algorithm to track the minima present among speech frames.

4.3. Parameters

In the minimum statistics based approach, the computational load is impacted most by the resolution of STFT and the interval over which the minimum is computed. Experiments reported in section 5 uses STFTs with a frequency resolution of 256 bins at a frame rate of 10ms. The minimum statistics are searched over an interval of 1.4 seconds of past speech. Our implementation took advantage of an efficient search algorithm described in [12], where the search interval is broken into sub-intervals (12 sub-intervals in our case) each of which is represented by its local minimum. The complete implementation runs in less than 2% real-time on a 1GHz Linux P-III machine. In addition to these parameters, the parameters for smoothing STFT and computing variances are identical to those used in [12].

4.4. Determination of Utterance Boundary

Once the background noise spectrum, $\hat{N}(k,t)$, is estimated, each input frame is classified as speech or non-speech. If the input, $Y(k,t)$, is higher than $\gamma\hat{N}(k,t)$ in at least one fifth of the spectral bins, it is classified as speech. To avoid triggering the speech detector due to noise perturbations, the constant is set to $\gamma = \sqrt{2}$. Finally, a simple state machine decides the utterance boundaries. Arrival of at least four consecutive speech frames marks the beginning of an utterance in the state machine. If no speech frame is present for the next 40 frames, the state machine transitions to mark the end of utterance. These utterance markers are expanded by padding six additional frames, before and after the utterance.

The latency of the algorithm largely depends on how the end of a segment is defined. Here, a segment is considered to have ended if it is followed by 400ms of non-speech. By reducing the lookahead buffer for determining the end of segment, the latency can be reduced proportionally. It may be possible to use certain spectral cues to determine the end of an utterance or a segment. Further, the degrees of freedom provided by the STFT domain can be used more effectively to remove new noise types that can be localized in STFT domain.

5. EXPERIMENTS & RESULTS

The new segmentation algorithm based on minimum statistics was tested on two significantly different tasks – (a) SPINE-2 test set, and (b) a subset of HMIHY task, an internal database of real customer interactions [1].

The SPINE task is a collection of speech from pairs of speakers participating in a cooperative task. Participants were placed in separate sound booths and subjected to one of the many background noise types, which included noise from F16, E3A AWACS, helicopter, armored vehicle such as bradley and hummer. The SPINE-2 tests contain about an hour and three hours of speech in development (SPINE-2 Dev) and evaluation (SPINE-2 Eval) set respectively. The evaluation test set has eight different background noise conditions, four of which are absent in the training data. The input consists of a continuous signal from a conversation-side. A state-of-the-art system was developed for testing on the SPINE task. It uses PLP input features, class-based trigram language model and applies a variety of compensations used in research systems – cepstral mean normalization, cepstral variance normalization, vocal tract normalization, constrained model-space adaptation, speaker-adapted training, and maximum likelihood linear regression based speaker adaptation. In addition, the input features were transformed using linear discriminant analysis and the transformation associated with semi-tied covariance approximation.

The HMIHY task contains response from customer to the open-ended prompt, “This is AT&T, how may I help you?”. Compared to the SPINE task, the HMIHY subset has lower amounts of background noise, but has larger variations in signal power. The subset was chosen to have at least 4s long response from users, and to contain at least one non-speech event as marked by transcribers. It contained noise from typical home environments as well as telephone tones. This subset was tested using currently deployed Watson system for the HMIHY task [1].

Table 2 shows comparison of three kinds of segmentations on both the domains. On the SPINE task, it includes the human labeled segmentation provided by the Linguistic Data Consortium (LDC-seg), output of a batch mode algorithm (Batch-seg, described in [9]), and the newly developed extreme statistics based algorithm (XStats-seg). On the HMIHY task, three segmentations include no segmentation of user response (No-seg), segmentation using an existing segmenter in the Watson system (Watson-seg, based on [13]), and the new segmentation.

SPINE-2 Dev	Mts. of Speech	WER
LDC-seg	77	24.3%
Batch-seg	76	27.5%
XStats-seg	73	24.9%

SPINE-2 Eval	Mts. of Speech	WER
LDC-seg	167	32.7%
Batch-seg	155	36.3%
XStats-seg	153	34.7%

HMIHY Task	Mts. of Speech	WER
No-seg	261	32.8%
Watson-seg	183	33.5%
XStats-seg	174	32.8%

Table 2. Performance comparison of segmentation algorithms.

As shown in table 2, on the SPINE-2 development test set, the new algorithm does almost as well as the LDC segments. On the evaluation test set, it does better than the current batch-mode algorithm and the LDC segments give the best results. In the HMIHY task, the new algorithm does significantly better than the currently deployed Watson segmentation algorithm, and the WER in this low noise task is almost the same as having no segmentation of the user responses. One advantage of making the speech / non-speech decision in the STFT domain is that the events such as telephone tones are highly localized and are easily filtered using a simple measure of its spectral spread. In all the tasks listed in table 2, the new algorithm appears to have reduced the number of unnecessary frames sent to the ASR system while providing lower WER than the current task-specific segmentation algorithms.

6. CONCLUSIONS & DISCUSSION

In this paper, we provide a solution for robust speech detection which can be applied across tasks without requiring task-specific tuning. We have demonstrated its effectiveness with tests on two tasks that differ in speaking style, noise type and bandwidth. On the SPINE-2 test set, which contains a variety of background noise, the new algorithm obtains lower WER than our current batch mode algorithm by about 2% WER, while sending fewer frames to the ASR decoder. On the SPINE-2 development set, the WER is almost equivalent to that using the human-derived segmentations from the LDC. On the HMIHY task, where the noise power is lower than the SPINE task, the new algorithm reduces WER by 0.7% over the currently deployed segmentation algorithm. The major advantage of the new algorithm in this task was that it reduced computational load associated with ASR decoding by sending 34% fewer frames to the decoder for processing.

7. ACKNOWLEDGMENTS

The authors would like to thank Chris Vanderveer for hand-correcting a subset of the SPINE word-level segmentations, and Anne Kirkland for providing Watson baseline system. We thank Hong Kook Kim for making his segmentations available for comparisons and for useful discussions. We also thank Michiel Bacchiani and Michael Riley for making their ASR software tools available.

8. REFERENCES

[1] Allen L. Gorin, Giuseppe Riccardi, and Jerry H. Wright, “”How may I help you”,” *Speech Communications*, vol. 23, pp. 113–127, 1997.

[2] Venkata R. Gadde, Andreas Stolcke, Dimitra Vergyri, Jing Zheng, Kemal Sonmez, and Anand Venkatraman, “Building an ASR system for noisy environment: SRI’s 2001 SPINE evaluation system,” *Proc. Int’l Conf. on Spoken Language Processing (ICSLP)*, 2002.

[3] Brian Kingsbury, George Saon, Lidia Mangu, Mukund Padmanabhan, and Ruhi Sarikaya, “Robust speech recognition in noisy environments: The 2001 IBM SPINE evaluation system,” *Proc. Int’l Conf. on Acoustic, Speech and Signal Processing*, 2002.

[4] “ITU-T Recommendation G.729/Annex A – speech coding algorithm: A silence compression scheme,” *ITU-T*, 1996.

[5] Qi Li, Jinsong Zheng, Augustine Tsai, and Qiru Zhou, “Robust endpoint detection and energy normalization for real-time speech and speaker recognition,” *IEEE Trans. on Speech and Audio Processing*, vol. 10, pp. 146–157, 2002.

[6] Murat Saraclar, Michael Riley, Enrico Bocchieri, and Vincent Goffin, “Towards automatic closed captioning: low latency real time broascast news transcription,” *Proc. Int’l Conf. on Spoken Language Processing (ICSLP)*, 2002.

[7] J. C. Spohrer, P. F. Brown, P. H. Hochschild, and J. K. Baker, “Partial backtrace in continuous speech recognition,” *Proc. Int’l Conf. on Systems, Man, and Cybernetics*, pp. 36–42, 1980.

[8] Tom Mitchell, *Machine Learning*, McGraw-Hill, 1997.

[9] Hong Kook Kim and Richard Rose, “Evaluation of robust speech recognition algorithms for distributed speech recognition in a noisy automobile environment,” *Proc. Int’l Conf. on Spoken Language Processing (ICSLP)*, pp. 233–237, 2002.

[10] Rolf-Dieter Reiss, *Approximate Distributions of Order Statistics (Chapters 5,6,9)*, Springer Series in Statistics, 1989.

[11] Emil J. Gumbel, *Statistics of Extreme (Chapter 7)*, Columbia University Press, 1958.

[12] Rainer Martin, “Noise power spectral density estimation based on optimal smoothing and minimum statistics,” *IEEE Trans. on Speech and Audio Processing*, vol. 9, no. 5, pp. 504–512, 2001.

[13] David Malah, Richard V. Cox, and Anthony J. Accardi, “Tracking speech-presence uncertainty to improve speech enhancement in non-stationary noise environments,” *Proc. Int’l Conf. on Acoustic, Speech and Signal Processing*, pp. 789–792, 1999.