Accurate Endpointing with Expected Pause Duration

Baiyang Liu, Bjorn Hoffmeister, Ariya Rastrow

Amazon, USA

{baiyangl, bjornh, arastrow}@amazon.com

Abstract

In an online automatic speech recognition system, the role of the endpoint detector is to infer when a user has finished speaking a query. Accurate and low-latency endpoint detection is crucial for natural voice interaction. Classic voice activity detector (VAD) based approaches monitor the incoming audio and trigger when a sufficiently long pause is detected. Such approaches are typically limited due to their inability to distinguish between within and end-of-sentence pauses. In this paper, we propose an endpoint detection algorithm that is integrated with the speech recognition process, leveraging acoustic and language model information in order to distinguish between within and end-of-sentence pauses. Unlike other integrated approaches that are based on the highest-scoring active recognition hypothesis, the proposed algorithm computes the expected pause duration over all active hypotheses, which leads to a more reliable pause duration prediction. We show that our method achieves significantly higher accuracy and lower latency in a comparison to standard approaches for endpoint detection.

Index Terms: Voice Activity Detection, Speech Endpointing, Speech Recognition

1. Introduction

In this paper we investigate endpoint detection algorithms for automatic speech recognition, which predict when the user has fully articulated an utterance. The exact definition of the end of an utterance depends on the application, typical examples include questions asked to a voice interface, commands given to a voice interface, or the answers to questions asked by an interactive system. In this study we assume that the onset of the decoding is initiated by the user or the system, for example by pressing a button. The automatic speech recognizer (ASR) decodes the audio stream until the ASR internal endpoint detector triggers the end of utterance. The endpoint detection problem is difficult mainly due to the following challenges:

1. Challenging acoustic conditions, where noise can be confused with the speech signal. Examples are environments with low signal-to-noise ratio (SNR), background chatter and side-talk.
2. Long hesitations within an utterance. Long within-utterance pauses will lead to an early endpoint and generate an incomplete utterance due to cutting off the remaining speech. Without semantic context information, it is challenging to predict whether more speech is expected while maintaining a low latency for detecting the end of utterance.

The classic voice-activity-detector (VAD) based endpoint detector scans the audio signal for non-speech regions. An endpoint will be detected when a non-speech region duration is longer than a given threshold. The speech/non-speech classification is often based on specifically engineered acoustic features, like audio energy [1], pitch [2], and zero-crossing rate [3]. The long-term spectral flatness measure (LSFM) was proposed for VAD under noisy environment at 10dB and lower SNR in [4]. The speech/non-speech information was also proposed to adjust the recognition hypothesis score in [5]. Privacy sensitive features were proposed for speech/non speech classification in [6].

In [7] , cortical features are derived from the posterior output of a DNN and the speech/non-speech classes are modeled using Gaussian mixture models (GMM). Recently, DNNs [8] and recurrent neural networks (RNNs) [9] have demonstrated superior VAD performance. Instead of detecting the segment based on the audio signal directly, [10] performs a full decoding on the unsegmented audio stream and detects non-speech regions in a post-processing step using the decoder alignment.

Besides the problem of non-speech region detection, another challenge is to differentiate the within-sentence pauses from end-of-sentence pauses. Prosodic and semantic based features have been proposed for pause type classification in [11, 12, 13]. In [11], the endpoint detection threshold is adjusted online based on a decision tree. An approach for combining a prosody analysis with the sentence end probabilities from an N-gram language model was proposed in [13] for distinguishing between within-sentence and end-of-sentence pauses.

All the above methods estimate the non-speech duration based on either the frame-level speech/non-speech classification result or based on the alignment from the 1-best recognition hypothesis. For stream processing, detecting the non-speech regions based on a frame-by-frame classification or from a single decoding hypothesis is not always robust as our empirical studies show. In this paper, we propose to use the "expected pause duration" for implementing an end-of-utterance detector. The expected pause duration is defined as follows: If we pick a single hypothesis from an ongoing decoding run, then the pause duration is simply the number of consecutive frames for which the hypothesis has been in a non-speech state, or zero if the hypothesis is currently in a speech state. By assigning a probability to the hypothesis and computing the weighted average over all active hypotheses, we derive the expected pause duration. We further define the "expected pause duration at sentence end" by calculating the expectation only over hypotheses in a language model end state. We will show that in combination with a simple thresholding approach, the expected pause duration based method outperforms the standard approaches to end-of-utterance detection.

Standard VAD-based approaches are introduced in section 2. In section 3, we describe in detail the proposed endpoint detection algorithm using the expected pause duration. Experimental setup, comparative results, and an analysis are discussed in section 4. Section 5 draws conclusions.

2. VAD-based Endpointing

This section introduces two standard VAD-based endpoint detection algorithms. These endpoint detectors usually work on the frame level and label each frame either as speech or non-speech frame. An end of utterance will be detected when certain amount of non-speech frames are observed in a sliding window.

2.1. Energy-based VAD

Energy-based voice activity detection labels each audio frame as speech or non-speech by thresholding the band-pass filtered energy. To compensate for changes in the acoustic environment, an energy range tracker is initialized with the energy level from the first couple of frames and is then continuously updated online. A label switching from speech frame to non-speech frame requires the energy of the current frame to be smaller than the low threshold, and switching from non-speech frame to speech frame requires the energy to be higher than high threshold of the range tracker.

The energy-based VAD are both simple and computational efficient. However, it is not robust to noisy environments with low SNR. For speech recognition under challenging environments, the energy of speech can be very dynamic and it is difficult to model. Energy-based VAD systems will generate higher early endpoint rates or missing endpoint rates in these cases.

2.2. DNN-based VAD

Deep neural networks have been widely used in ASR for acoustic modeling and shown to outperform GMM [14]. A DNN model with two outputs can be trained using a large corpus of speech/non-speech labels. Each audio frame is classified as speech/non-speech using likelihood ratio test. The model-based frame classification is more robust to noisy and low SNR environments than the energy-based VAD.

3. Decoder Integrated Endpointing

The speech recognizer processes the audio stream and tries to detect an end-of-utterance at each frame based on features derived from the search space. Once an endpoint is detected, the recognizer generates and returns the recognition result.

For achieving a high endpoint detection accuracy and low latency, it is necessary to differentiate between the within-sentence pause and end-of-sentence pause. However, it is challenging to classify them deterministically. For example, "What is the weather" is a valid complete sentence. Users could also add additional information after some pause, like "What is the weather [Pause] in Seattle", or "What is the weather [Pause] in Seattle [Pause] tomorrow". In an HMM-based ASR system, the end-of-sentence probability is modeled by the end states of the language model. An endpoint detector integrated in the decoder can take advantage of the information. In this section we discuss two integrated endpoint detection solutions, the first one based on the best active decoding hypothesis and the second one based on our newly proposed expected pause duration.

3.1. Decoder Search Space

For each audio frame, the decoder search space is expanded based on the given decoding graph, which includes both acoustic model (AM) score and language model (LM) score. The acoustic and language scores are accumulated along the decoding path. Let $X_t = \{x_1, x_2, x_3..., x_t\}$ be the sequence of audio frames until t, and let $S_t^i = \{s_1^i, s_2^i, s_3^i, ..., s_t^i\}, i = [1, N_t]$ be the state sequence of the ith active hypothesis at time t. For any given time t, N_t is the number of active hypotheses. The posterior of the hypothesis can be presented as:

$$P(S_t^i|X_t) = \frac{P(s_t^i|x_t)}{P(s_t^i)} P(s_t^i|s_{t-1}^i) P(S_{t-1}^i|X_{t-1}) \quad (1)$$

where $P(s_t^i|x_t)/P(s_t^i)$ is the acoustic score using the posterior generated by the deep neural network and normalized by the state prior. $P(s_t^i|s_{t-1}^i)$ is the multiplication of language model probabilities and HMM transition probabilities. By L_t^i we denote the pause duration for the i-th hypothesis. Formally, we can define L_t^i as the largest integer N such that $s_{t-N+1}^i \in S_{\text{NS}} \wedge \cdots \wedge s_t^i \in S_{\text{NS}}$ holds, where S_{NS} denotes the set of all non-speech states.

3.2. Endpoint Detection based on 1-Best Pause

The pause duration L_t^i for a given hypothesis i can be derived from the traceback through the decoding graph kept by the decoder. The input label of each arc in the decoding graph can be mapped to an HMM state representing a context dependent phone, and hence can be further mapped to the speech or non-speech case.

For making the endpoint decision we consider only hypotheses being in a language model end state, and among these hypotheses we select the best scoring one. The endpoint detector triggers if the selected hypothesis is the overall best hypothesis,

$$\arg\max_{i, s_t^i \in S_{\text{end}}} P(S_t^i|X_t) = \arg\max_i P(S_t^i|X_t), \quad (2)$$

and its pause duration exceeds a given threshold,

$$L_t^i > T_{\text{end}} \quad \text{with} \quad i := \arg\max_{i, s_t^i \in S_{\text{end}}} P(S_t^i|X_t), \quad (3)$$

where T_{end} is a tunable threshold. An edge case, which the algorithm doesn't handle gracefully, can arise when the language model fails to correctly predict the end of the sentence. In that case the probability of the best non-sentence end hypothesis continues to prevail and the endpoint detector doesn't trigger. We can solve the problem by alternatively triggering if the pause duration of the overall best hypothesis exceeds a threshold,

$$L_t^i > T \text{ with } i := \arg\max_i P(S_t^i|X_t), \quad (4)$$

where we choose T such that $T > T_{\text{end}}$.

A problem we observe is that the algorithm is not always robust due to the fact that the locally best hypothesis is not necessarily part of the end result and can even change rapidly between frames; this is mainly due to two reasons inherent to an HMM-based ASR decoder:

- The language model "corrects" the hypothesis throughout the decoding run.
- As an artifact of the decoding graph optimization the language model score distribution along a path in the decoding graph is not smooth, but can take the form of a step function.

Instead of using the locally best hypothesis, we propose to take all active hypotheses into consideration, which leads to the expected pause based endpoint detector.

3.3. Endpoint Detection based on Expected Pause

Let L_t^i be the pause duration of the i-th hypothesis, then the expected pause duration at time t is defined as

$$\mathbb{D}(L_t) := \sum_i L_t^i P(S_t^i|X_t), \tag{5}$$

which can be interpreted as an expectation of the pause duration computed over all active decoding hypotheses. Smoothing the pause duration by considering all hypotheses makes the value less sensitive to changes in the best decoding hypothesis.

We take the sentence end information into account by modifying the above equation to only consider hypotheses being in a language model end state:

$$\mathbb{D}_{\text{end}}(L_t) := \sum_{i, s_t^i \in S_{\text{end}}} L_t^i P(S_t^i|X_t), \tag{6}$$

Obviously, the relation $\mathbb{D}(L_t) \geq \mathbb{D}_{\text{end}}(L_t)$ holds with equality being achieved only if all active hypotheses are in an end state. For within-sentence pauses the value of $\mathbb{D}_{\text{end}}(L_t)$ will stay small, while it will converge to $\mathbb{D}(L_t)$ for a pause at the end of a sentence.

The decision making process is again a simple thresholding approach, where we trigger if either of the two conditions is met:

1. $\mathbb{D}_{\text{end}}(L_t) > T_{\text{end}}$ and $\mathbb{D}(L_t) > T'$, or
2. $\mathbb{D}(L_t) > T$

The additional $\mathbb{D}(L_t) > T'$ term in the first condition is a safety against triggering too early in some edge cases, which we observed in our data. An example for such an edge case is the sentence "What is the weather [Pause] in Seattle?", where "[Pause]" might have a fifty-fifty chance to be a within-sentence or end-of-sentence pause. The chance of triggering prematurely in such an edge case increases with tuning T_{end} aggressively for low latency. The second condition is to safe guard against the case of the language model failing to correctly predict the end of sentence; we trigger also at a sufficiently long within-sentence pause.

A problem we observe with the expected pause duration are many false triggers at the onset of an utterance. Before and at the onset of speech, the empty utterance is a likely hypothesis and hence our expected pause based endpoint detector triggers. We currently avoid the problem by starting with a simple VAD and switch to the expected pause endpointer after detecting a certain number of speech frames.

4. Experiments

This section presents comparative results of the endpoint detectors introduced in the previous sections. We use the evaluation set from an in-house data collection to report results. The metrics for evaluating the endpoint detector performance are early and missed endpoint rate (EEPR and MEPR), as well as latency for a successful endpoint detection. We also compare word error rate (WER) and sentence error rate (SER), which are directly impacted by the endpoint detector performance. The overall comparative results show that the proposed endpoint detection algorithm achieves the lowest error rates at low latency.

4.1. Experimental Setup

For the experiments, we use an internally collected data set with an evaluation set that is twice as large as the development set.

Table 1: Endpoint detection performance on the evaluation set. All numbers are relative changes to the baseline defined by the energy-VAD.

	WER	SER	EEPR	MEPR	latency
energy-VAD	1.0	1.0	1.0	1.0	1.0
1best-pause	0.96	0.95	0.61	0.79	1.32
expected-pau	0.95	0.94	0.55	0.57	1.02

The development set is used to tune the thresholds of the endpoint detectors; all numbers are reported on the evaluation set. In the experiments we assume the startpoint to be given and we only focus on the endpoint detection performance. The reference timestamps of the sentence end is obtained by running a forced-alignment of the transcription against the audio data.

We compare an energy-VAD, with a bandpass filter applied to the audio signal for filtering out non-speech relevant frequencies, and the two decoder integrated endpoint detection algorithms described in Section 3. The underlying speech recognition system is an DNN-HMM based large vocabulary system.

4.2. Endpoint Detection Performance Metrics

The endpoint detection accuracy and latency are measured as follows:

- Word Error Rate (WER)
- Sentence Error Rate (SER)
- Early Endpoint Rate (EEPR): An early endpoint event happens if the endpoint detector triggers before the end of utterance is reached. An early endpoint usually results in cutting off speech.
- Missed Endpoint Rate (MEPR): In this work we count a missed endpoint event for any utterance for which the end point detector did not trigger within two seconds of end of utterance. Or in other words, a latency larger than two seconds is considered as missed endpoint. Missed endpoints manifest either in an unacceptable long latency or in hypothesizing additional words from picking up on noise or background speech.
- Endpoint Latency: The endpoint latency is the gap between the end of utterance and the moment the endpoint detector triggers. We report the latency for a data set as the median over all utterances in the data set, not including early or missed endpoints. In a realtime application latency is an important metric strongly correlated to the perceived "naturalness" of the endpointer solution.

We report results as relative changes in WER, SER, EEPR, MEPR, and latency compared to a baseline provided by the energy-VAD endpointer.

4.3. Results

In Figure 1, the early endpoint rates on the development set for the three algorithms are plotted by sweeping over the parameters of the endpointer algorithms. As expected, the early endpoint rates decrease with an increase in latency. The proposed expected pause duration based endpoint detector generates the lowest early endpoint rate over all latency ranges.

In Table 1, the final results on the evaluation data set are presented. The operating points for the three endpoint detectors are chosen to give an optimal trade-off between EEPR,

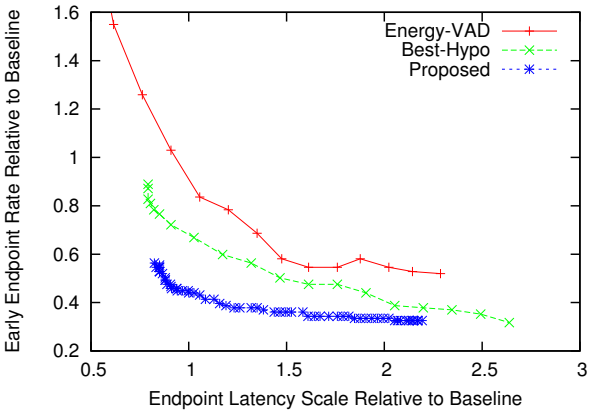

Figure 1: Early Endpoint Rate vs. Endpoint Latency on the development set. All numbers are relative changes to the baseline defined by the energy-VAD.

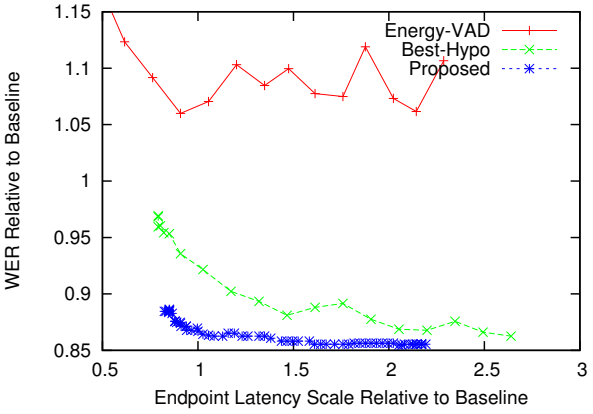

Figure 2: WER vs. Endpoint Latency on the evaluation set. All numbers are relative changes to the baseline defined by the energy-VAD.

MEPR, and latency. The 1-best-pause algorithm outperforms the energy-VAD clearly on EEPR and MEPR, but pays with a higher median latency. However, Figure 1 indicates that the 1-best-pause endpoint detector would still outperform the energy-VAD on equal latency. The proposed endpoint detection algorithm using the expected pause clearly outperforms both competitors and achieves optimal EEPR and MEPR already at a low median latency. The reductions in word and sentence error rates are correlated with the reductions in EEPR and MEPR and show the importance of accurate endpoint detection for speech recognition performance.

For realtime applications we are usually interested in reducing the latency to a In Figure 2 we explore the dependency of WER on latency for the three endpoint detection algorithms. The figure shows how little robust the energy-VAD is compared to the two decoder-integrated endpoint detectors, and the expected pause duration approach clearly outperforming the other two endpointers. The curve for the expected pause duration based endpointer shows the nicest properties, smooth and quickly, i.e., with low latency, reaching the region of optimal WER.

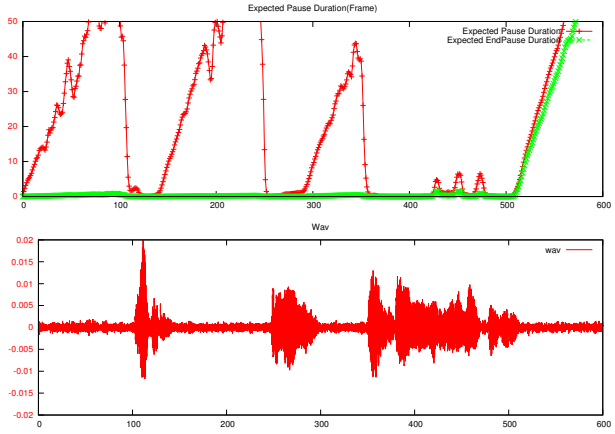

Figure 3: Plot of the expected overall pause duration (red line) and the expected end-of-sentence pause duration (green line) for a given speech sample.

4.4. Example

An example of the expected pause duration and expected end pause duration is presented at Figure 3. The speech signal shows clearly how the speaker pauses after the first and also after the second word before finishing the sentence. The probability of being end state after the first word is small in the language model and hence the expected end-of-sentence pause duration after the first word is small (green line in figure 3). In contrast, the overall expected pause duration peaks after the first word (red line). The same pattern can be observed after the second word, but changes at the end of the sentence. Now, the language model probability of being in an end state is high and so is the expected end-of-sentence pause duration (peaking green line). We can safely conclude that an endpoint is detected after a short wait at the end of the sentence.

5. Conclusion

Speech endpoint detection is an important component of an automatic speech recognition system, solving the difficult task of detecting when a user finished speaking an utterance. We introduced a new, speech decoder integrated endpoint detection algorithm, which combines acoustic and lexical clues from the speech decoder for computing an expected end-of-sentence pause duration. The proposed algorithm outperforms a simple energy based VAD and compares also favorably to an endpoint detection algorithm based on the highest scoring hypothesis of the speech decoder.

A detailed analysis shows that the expected pause duration approach performs in particular well under low latency constraints, a requirement typically found in realtime applications.

6. References

[1] K.-H. Woo, T.-Y. Yang, K.-J. Park, and C. Lee, "Robust voice activity detection algorithm for estimating noise spectrum," *Electronics Letters*, vol. 36, no. 2, pp. 180–181, 2000.

[2] R. Chengalvarayan, "Robust energy normalization using speech/nonspeech discriminator for german connected digit recognition," in *Sixth European Conference on Speech Communication and Technology*, 1999.

[3] A. ITU, "silence compression scheme for g. 729 optimized for

terminals conforming to recommendation v. 70," *ITU-T Recommendation G*, vol. 729, 1996.

[4] J. Ramırez, J. C. Segura, C. Benıtez, A. De La Torre, and A. Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech communication*, vol. 42, no. 3, pp. 271–287, 2004.

[5] T. Oonishi, P. R. Dixon, K. Iwano, and S. Furu, "Robust speech recognition using vad-measure-embedded decoder," *INTERSPEECH*, pp. 2239–2242, 2009.

[6] S. H. K. Parthasarathi, D. Gatica-Perez, H. Bourlard, and M. Magimai-Doss, "Privacy-sensitive audio features for speech/nonspeech detection," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 8, pp. 2538–2551, 2011.

[7] S. Thomas, S. H. R. Mallidi, T. Janu, H. Hermansky, N. Mesgarani, X. Zhou, S. A. Shamma, T. Ng, B. Zhang, L. Nguyen, and S. Matsoukas, "Acoustic and data-driven features for robust speech activity detection." in *INTERSPEECH*, 2012.

[8] N. Ryant, M. Liberman, and J. Yuan, "Speech activity detection on youtube using deep neural networks." in *INTERSPEECH*, 2013, pp. 728–731.

[9] T. Hughes and K. Mierle, "Recurrent neural networks for voice activity detection," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7378–7382.

[10] S. Stüker, C. Fügen, F. Kraft, and M. Wölfel, "The isl 2007 english speech transcription system for european parliament speeches." in *INTERSPEECH*, 2007, pp. 2609–2612.

[11] A. Raux and M. Eskenazi, "Optimizing endpointing thresholds using dialogue features in a spoken dialogue system," in *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*. Association for Computational Linguistics, 2008, pp. 1–10.

[12] J. Edlund, M. Heldner, and J. Gustafson, "Utterance segmentation and turn-taking in spoken dialogue systems," *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen*, pp. 576–587, 2005.

[13] L. Ferrer, E. Shriberg, and A. Stolcke, "Is the speaker done yet? faster and more accurate end-of-utterance detection using prosody," in *Seventh International Conference on Spoken Language Processing*, 2002.

[14] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.