COMBINING ACOUSTIC EMBEDDINGS AND DECODING FEATURES FOR END-OF-UTTERANCE DETECTION IN REAL-TIME FAR-FIELD SPEECH RECOGNITION SYSTEMS

Roland Maas, Ariya Rastrow, Chengyuan Ma, Guitang Lan
Kyle Goehner, Gautam Tiwari, Shaun Joseph, Björn Hoffmeister

Amazon, USA

ABSTRACT

We present an end-of-utterance detector for real-time automatic speech recognition in far-field scenarios. The proposed system consists of three components: a long short-term memory (LSTM) neural network trained on acoustic features, an LSTM trained on 1-best recognition hypotheses of the automatic speech recognition (ASR) decoder, and a feed-forward deep neural network (DNN) combining embeddings derived from both LSTMs with pause duration features from the ASR decoder. At inference time, lower and upper latency (pause duration) bounds act as safeguards. Within the latency bounds, the utterance end-point is triggered as soon as the DNN posterior reaches a tuned threshold. Our experimental evaluation is carried out on real recordings of natural human interactions with voice-controlled far-field devices. We show that the acoustic embeddings are the single most powerful feature and particularly suitable for cross-lingual applications. We furthermore show the benefit of ASR decoder features, especially as a low cost alternative to ASR hypothesis embeddings.

***Index Terms*—** end-pointing, end-of-query detection, turn taking, dialog modeling, online speech recognition

1. INTRODUCTION

Consumer products with voice-enabled interfaces are on the rise. Popular examples comprise smartphones, navigation devices, and personal assistants, such as the Amazon Echo or Google Home. Typically, the interaction with such devices is user initiated, e.g., by pressing a button or uttering a wake-word. In contrast, the end of an utterance is often to be inferred automatically by the ASR system. This task of end-point detection, or end-pointing, needs to address two major problems: late end-points due to background noise as well as early end-points in case of long within utterance pauses.

The classical end-pointing approaches are based on different types of voice-activity-detection (VAD) where the end-pointer would trigger if a non-speech region of a certain length is detected. While VAD systems can be based on engineered acoustic features [1–5], such as audio energy [6], pitch [7], zero-crossing rate [8, 9], and cortical features [10], superior performance is typically achieved with DNN-based approaches [11–14]. Besides detecting silence regions, it is furthermore important to differentiate within-sentence pauses from end-of-sentence pauses, which can be achieved by employing decoder [15, 16], acoustic and lexical features [17–21]. Recent publications showed convincing performance by training recurrent neural network variants, e.g., LSTMs, on acoustic and lexical features [22–26]. Specifically, in [22,26], an LSTM is trained directly on acoustic features to predict utterance end-points. In [25], time-asynchronous sequential networks, trained on acoustic and word embedding features, are stacked.

In this publication, we investigate a combination of LSTM-based classifiers, similarly to [25, 26], as follows: First, an LSTM is trained on acoustic features with frame-wise multi-task targets to predict both the utterance end-point as well as voice activity. Second, an LSTM is trained on embeddings of the 1-best ASR hypothesis. Third, a DNN is trained on frame-wise end-pointing targets combining three types of input features: the final layer representations of the acoustic and word LSTMs as well as pause duration estimates from the ASR decoder. At inference time, lower and upper latency (pause duration) bounds act as safeguards. Within the latency bounds, the utterance end-point is triggered as soon as the DNN posterior reaches a tuned threshold. The main contributions of this paper can be summarized as follows: We compare the performance of acoustic and ASR hypothesis embeddings (both obtained from LSTMs) on *real recordings from far-field devices*, showing that acoustic embeddings are the single most powerful feature. To the best of our knowledge, all related previously published studies have been carried out on close-talk or simulated data. We evidence that pause duration features from the ASR decoder provide an additional performance boost (when combined with acoustic embeddings) and may be a attractive alternative to ASR hypothesis embeddings (which can be costly to implement in real-time systems). Furthermore, we show that lower and upper pause duration thresholds are beneficial for controlling the end-pointing performance in far-field applications. We

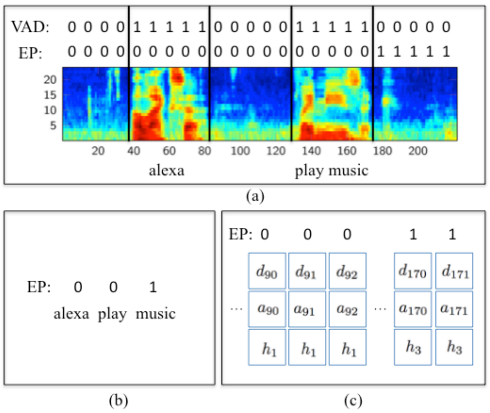

Fig. 1. Illustration of features, end-pointing (EP) targets, and VAD targets used to train the three components of the proposed end-point detector: the (a) acoustic LSTM, (b) word LSTM, and (c) classification layer.

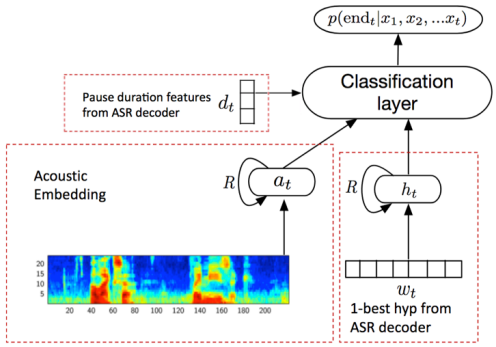

Fig. 2. Proposed end-point detector based on the combination of three features types: acoustic embeddings, ASR hypothesis embeddings, and decoder features.

also exemplify that acoustic features are particularly useful in cross-lingual settings, showing that acoustic embeddings obtained from English data yield convincing performance on German data.

2. SYSTEM ARCHITECTURE

In this section, we provide an overview of the proposed end-of-utterance detection system while drawing from various previous publications [16, 22, 25, 26].

2.1. Training

For the remainder of the paper, we refer to the LSTM trained on acoustic, i.e., LFBE, features as the *acoustic LSTM*. The LSTM trained on ASR hypotheses, i.e., word embeddings, is referred to as the *word LSTM*.

For training the acoustic LSTM, forced alignment is performed on transcribed utterances to derive binary frame-wise VAD and end-pointing targets, as in Fig. 1 a), similarly to [26]. The acoustic LSTM is then trained in a multi-task fashion on both target types. The VAD output posterior of the LSTM is used at inference time to measure pause duration in order to impose a lower latency bound. The end-pointing output posterior is not used at inference time. Instead, the hidden representation of the final LSTM layer (before the final affine/softmax transform) is used. We denote this representation as the *acoustic utterance embedding* a_t at frame t, cf. Fig. 2.

For the word LSTM training, we first decode the training corpus and convert the ASR hypotheses to pre-trained word embeddings vectors. The LSTM is then trained on binary end-pointing targets, as depicted in Fig. 1 b), similarly to [25]. The *ASR hypothesis embedding* vector at frame t is obtained from the hidden representation of the final LSTM layer and referred to as h_t, cf. Fig. 2.

In order to obtain information about the current decoder state, we derive pause duration estimates from the runtime decoding graph as proposed in [15, 16]: 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 is denoted by $p(S_t^i|X_t)$. 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 [15]. The *best path pause duration* is then given by $L_t^{i_{\max}}$ with $i_{\max} := \arg\max_i p(S_t^i|X_t)$. An estimate of the within-sentence pause length is represented by the *expected pause duration* $\mathbb{D}(L_t) := \sum_i L_t^i p(S_t^i|X_t)$. Furthermore, the end-of-sentence pause can be estimated by the *the expected final pause duration* $\mathbb{D}_{\text{end}}(L_t) := \sum_{i, s_t^i \in S_{\text{end}}} L_t^i p(S_t^i|X_t)$ with S_{end} denoting the set of end-states. The decoder feature vector at frame t is hence defined as $d_t = [L_t^{i_{\max}}, \mathbb{D}(L_t), \mathbb{D}_{\text{end}}(L_t)]$. As detailed in [16], the expected final pause duration $\mathbb{D}_{\text{end}}(L_t)$ correlates well with the probability mass of decoder tokens that are in a final HCLG state, hence, capturing the ASR decoder's belief on whether an utterance is completed. While this feature might provide partly redundant information compared to the word LSTM, the latter can potentially capture longer dependencies than an n-gram model (here used as G transducer).

Subsequently, we form joint feature vectors $f_t = [a_t, h_t, d_t]$ at every frame by concatenating the three feature types, as shown in Fig. 1 c): i) the hidden representations a_t of the last layer of the acoustic LSTM, ii) the hidden representation h_t of the last layer the word LSTM, iii) the decoder features d_t. Since the word LSTM does not run frame-synchronously, the same partial hypothesis embedding h_t is

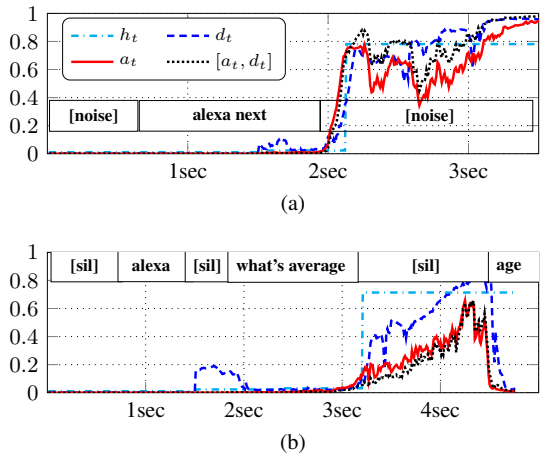

Fig. 3. Posterior probability plotted over time for end-point detectors that are trained on different feature types.

repeatedly appended as long as the 1-best decoding hypothesis remains unchanged. Finally, a joint classification layer, i.e., a fully connected DNN depicted in Fig. 2, is trained on the joint feature vectors f_t shown in Fig. 1 c).

2.2. Inference

At runtime, the end-pointer operates as follows: i) If a minimum pause duration $T_{\min}$ is not yet reached, no end-point is triggered. The pause duration is estimated by counting the number non-speech frames, as classified by the VAD output layer of the acoustic LSTM. ii) If a maximum pause duration $T_{\max}$ is reached, the end-point is enforced. This pause duration is estimated from the time $L_t^{i_{\max}}$ the 1-best ASR decoding hypothesis has been in a non-speech state. iii) Within the latency bounds, the utterance end-point is triggered as soon as the DNN posterior reaches a tuned threshold δ.

3. EXPERIMENTS

In this section, we turn to an experimental analysis of the proposed end-pointing system.

3.1. Experimental setup

As dataset, we use real recordings of natural human interactions with voice-controlled far-field devices. These recordings are not controlled in any way and may hence also contain multiple talkers and background noise at any position in the utterance. We consider two different corpora: English and German. The English training and test corpora consist of 300h and 5h, respectively. The German training and test corpora consist of 40h and 5h, respectively.

The input features for the acoustic LSTM are 64 dimensional LFBEs. Both the acoustic and word LSTM consist of two layers with 100 cells per layer. The classification layer is a feed-forward DNN with two layers and 100 neurons per layer.

features	WERR	EEPR	P50	P90	P99
$[T_{\min} = 400, T_{\max} = 1500]$					
$[d_t]$	–	–	380	720	1500
$[h_t]$	-17%	-68%	380	1500	1510
$[a_t]$	-11%	-43%	380	750	1500
$[a_t, d_t]$	-15%	-54%	370	730	1500
$[a_t, h_t]$	-16%	-61%	360	760	1500
$[a_t, h_t, d_t]$	-16%	-59%	360	720	1500
$[T_{\min} = 400, T_{\max} = \inf]$					
$[a_t, h_t, d_t]$	-18%	-72%	360	690	2140
$[T_{\min} = 200, T_{\max} = 1500]$					
$[a_t, h_t, d_t]$	-14%	-53%	180	700	1500
$[T_{\min} = 0, T_{\max} = 1500]$					
$[a_t, h_t, d_t]$	-14%	-51%	50	680	1500

Table 1. End-point detection performance on English test data. The vector "a_t" refers to the acoustic embeddings (obtained from the acoustic LSTM), "h_t" refers to the hypothesis embeddings (obtained from the word LSTM trained on ASR hypotheses), and "d_t" refers to the decoder pause duration features. All duration/latency values are in msec.

We evaluate the ASR and end-pointing performance using the following metrics: a) relative Word Error Rate Reduction (WERR), b) Early Endpoint Rate (EEPR) describing how often the end-point detector triggers before the end of the last word in an utterance is reached (which usually results in cutting off speech), c) end-pointing latency in msec at P50, P90, and P99 quantiles describing the gap between the end of utterance and the moment the end-point detector triggers.

In order to allow for a meaningful WER and EEPR comparison, we tuned the end-pointing thresholds δ such that all investigated end-point detectors approximately operate at the same P50 and P90 latency.

3.2. Experimental results for different feature types

To obtain a better intuition for the investigated features, we train the final classification layer in Fig. 2 separately on decoder features d_t, ASR hypotheses h_t, and acoustic embeddings a_t and plot the posteriors over time for two example utterances, as shown in Fig. 3. The classifiers trained on acoustic embeddings a_t or decoder features d_t exhibit a (more or less) monotonic increase in posterior probability over time

during a speech pause. Considering that the posterior of the a_t classifier raises similarly to the posterior of d_t classifier, we can conclude that the acoustic LSTM implicitly learns a notion of "pause duration" (we can speculate that one set of neurons is responsible for counting non-speech frames). In Fig. 3 (a), the a_t posterior quickly reaches a value of 0.7 (which turned out to be the optimum tuned threshold for this classifier: $\delta = 0.7$) while in Fig. 3 (b), the a_t posterior remains below 0.7, correctly indicating a non-final pause. The acoustic LSTM, hence, seems to learn both notions of "finality" as well as length of a pause. We can observe that the decoder feature d_t classifier exhibits essentially two types of behaviors (which we verified by inspecting many more examples): If the language model (more precisely, the HCLG) is in end-state, the increase in posterior is rather steep, as in Fig. 3 (a). In contrast, if the decoder is not in end-state, the posterior value increases almost linearly with increasing pause duration, as in Fig. 3 (b). We furthermore found the decoder features to be fairly robust to interfering speech as it leverages both the acoustic and the language model to make speech/noise decision. This might explain why the $[a_t, d_t]$ classifier shows the most desirable behavior in the two depicted examples: In Fig. 3 (a), the $[a_t, d_t]$ classifier appears to be more noise robust than the a_t classifier. In Fig. 3 (b), the $[a_t, d_t]$ classifier exhibits a slower increase during the non-final pause than the a_t classifier (probably because the decoder signal indicating that the language model is not in end-state provides complementary information to a_t). The classifier trained on hypothesis embeddings yields a step function shape as the input only changes as the ASR 1-best hypothesis changes. It hence does not take the pause duration into account for the end-pointing decision.

Turning to a quantitative analysis, Table 1 compares the end-pointing performance for different features combinations as well as different values of the lower/upper safeguard thresholds. As evidenced above, the h_t (ASR hypothesis) classifier is a "hit-or-miss" approach: if it incorrectly classifies an utterance embedding as non-final, it cannot recover from it, even as the pause duration is increasing, therefore yielding unacceptably high latency (the P90 of 1.5sec shows that the upper safeguard threshold is often met). The best performance is achieved with the acoustic embedding (among the single feature classifiers). The combination of acoustic embeddings and hypothesis embeddings (with or without decoder features) yield the best overall performance. However, the implementation of hypothesis embedding features in a real-time ASR system can substantially increase the overall system complexity, which may or may not be acceptable. In cases where the implementation and/or run-time costs are undesirable, the combination of acoustic and decoder features appears to be a surprisingly attractive alternative, closely matching the performance of the $[a_t, h_t, d_t]$ classifier. Table 1 shows that increasing $T_{\max}$ to ∞ can cause end-points to entirely slip (P99 of 2sec indicates that the end of the audio file is met). Reducing the lower threshold $T_{\min}$ below 400msec negatively affects the performance due to an increase in early end-point rate.

a_t LSTM training	class. layer training	WERR	EEPR	P50	P90	P99
EN	EN	–	–	420	590	1500
EN	DE	-6%	-14%	420	580	1500
EN+DE	EN+DE	-6%	-15%	420	580	1500
DE	DE	-9%	-20%	420	580	1500

Table 2. End-point detection performance on German test data with English (EN) and German (DE) training data. In this experiments, only acoustic embeddings a_t are used for end-pointing detection. All duration/latency values are in msec.

3.3. Experimental results for mixed languages

For practical applications, it can be of interest to apply end-point detectors in multi-lingual settings, e.g., in low-resource languages. Furthermore, certain applications may need to handle code switching, such as in voice messaging where the invocation phrase ("send a message to my brother in Germany") and the actual message payload ("Alles Gute zum Geburtstag!") may be uttered in different languages. Table 2 compares the performance of acoustic LSTMs on German test data. It can be observed that the LSTM trained on multi-lingual (EN+DE) data performs only slightly worse than the LSTM that is exclusively trained on German (DE) data. Adapting only the classification layer to German data appears to have a similar effect as multi-lingual training.

Irrespective of the language aspect, we generally found that the performance of the acoustic LSTM is influenced by the variety of training data. If it is only presented with short command-style utterances at training, it may perform poorly on long dictation-type utterances (yielding early end-points) or confirmation utterances (yielding late end-points). However, if new use cases (types of utterances) are encountered, adapting the separate classification layer, in Fig. 2, on an according data set may offer an efficient alternative to retraining the entire LSTM.

4. CONCLUSIONS

We proposed an end-of-utterance detection system for real-time speech recognition in far-field scenarios and showed that ASR decoder features and acoustic embeddings, obtained from LSTMs, are a particularly useful features for practical applications. The proposed architecture, wherein a separate classification layer is employed, allows for resource-efficient adaptation to new domains and across languages.

5. REFERENCES

[1] W.-H. Shin, B.-S. Lee, Y.-K. Lee, and J.-S. Lee, "Speech/non-speech classification using multiple features for robust endpoint detection," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, 2000, pp. 1399–1402.

[2] T. T. Kristjansson, S. Deligne, and P. A. Olsen, "Voicing features for robust speech detection," in *Proceedings Interspeech*, 2005, pp. 369–372.

[3] X. Li, H. Liu, Y. Zheng, and B. Xu, "Robust speech endpoint detection based on improved adaptive band-partitioning spectral entropy," in *Proceedings of the Life System Modeling and Simulation International Conference on Bio-Inspired Computational Intelligence and Applications*, 2007, pp. 36–45.

[4] M. Fujimoto, K. Ishizuka, and T. Nakatani, "Study of integration of statistical model-based voice activity detection and noise suppression." in *Proceedings Interspeech*, 2008, pp. 2008–2011.

[5] S. O. Sadjadi and J. H. L. Hansen, "Unsupervised speech activity detection using voicing measures and perceptual spectral flux," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 197–200, 2013.

[6] 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.

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

[8] A. ITU, "Silence compression scheme for G. 729 optimized for terminals conforming to recommendation V. 70," *ITU-T Recommendation G*, vol. 729, 1996.

[9] L. Lu, H.-J. Zhang, and H. Jiang, "Content analysis for audio classification and segmentation," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 7, pp. 504–516, 2002.

[10] 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 *Proceedings Interspeech*, 2012.

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

[12] Xiao-Lei Zhang and Ji Wu, "Deep belief networks based voice activity detection," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 697–710, 2013.

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

[14] R. Maas, S. H. K. Parthasarathi, B. King, R. Huang, and B. Hoffmeister, "Anchored speech detection," in *Proceedings Interspeech*, 2016, pp. 2963–2967.

[15] B. Liu, B. Hoffmeister, and A. Rastrow, "Accurate endpointing with expected pause duration." in *Proceedings Interspeech*, 2015, pp. 2912–2916.

[16] R. Maas, A. Rastrow, K. Goehner, G. Tiwari, S. Joseph, and B. Hoffmeister, "Domain-specific utterance end-point detection for speech recognition," in *Proceedings Interspeech*, 2017, pp. 1943–1947.

[17] L. Ferrer, E. Shriberg, and A. Stolcke, "A prosody-based approach to end-of-utterance detection that does not require speech recognition," in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2003, pp. 605–608.

[18] H. Arsikere, E. Shriberg, and U. Ozertem, "Computationally-efficient endpointing features for natural spoken interaction with personal-assistant systems," in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 3241–3245.

[19] K. Audhkhasi, K. Kandhway, O. D. Deshmukh, and A. Verma, "Formant-based technique for automatic filled-pause detection in spontaneous spoken English," in *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 4857–4860.

[20] H. Arsikere, E. Shriberg, and U. Ozertem, "Enhanced end-of-turn detection for speech to a personal assistant," in *Proceedings AAAI Spring Symposium on Turn-taking and Coordination in Human-Machine Interaction*, 2015.

[21] Y. Ishimoto, T. Teraoka, and M. Enomoto, "End-of-utterance prediction by prosodic features and phrase-dependency structure in spontaneous japanese speech," in *Proceedings Interspeech*, 2017, pp. 1681–1685.

[22] M. Shannon, G. Simko, S. yiin Chang, and C. Parada, "Improved end-of-query detection for streaming speech recognition," in *Proceedings Interspeech*, 2017, pp. 1909–1913.

[23] A. Maier, J. Hough, and D. Schlangen, "Towards deep end-of-turn prediction for situated spoken dialogue systems," in *Proceedings Interspeech*, 2017, pp. 1676–1680.

[24] C. Liu, C. Ishi, and H. Ishiguro, "Turn-taking estimation model based on joint embedding of lexical and prosodic contents," in *Proceedings Interspeech*, 2017, pp. 1686–1690.

[25] R. Masumura, T. Asami, H. Masataki, R. Ishii, and R. Higashinaka, "Online end-of-turn detection from speech based on stacked time-asynchronous sequential networks," in *Proceedings Interspeech*, 2017, pp. 1661–1665.

[26] S. yiin Chang, B. Li, T. Sainath, G. Simko, and C. Parada, "Endpoint detection using grid long short-term memory networks for streaming speech recognition," in *Proceedings Interspeech*, 2017, pp. 3812–3816.