TOWARDS ACCURATE AND REAL-TIME END-OF-SPEECH ESTIMATION

Yifeng Fan[1] [*], *Colin Vaz*[2], *Di He*[2], *Jahn Heymann*[2], *Viet Anh Trinh*[2], *Zhe Zhang*[2], *Venkatesh Ravichandran*[2]

[1]University of Illinois at Urbana-Champaign
[2]Amazon Alexa

ABSTRACT

We introduce a variant of the endpoint (EP) detection problem in automatic speech recognition (ASR), which we call the end-of-speech (EOS) estimation. Given an utterance, EOS estimation aims to identify the timestamp when the utterance waveform has fully decayed and is then used to measure the EP latency. Accurate EOS estimation is difficult in large-scale streaming audio scenarios due to the hefty traffic and hardware limitations. To this end, we develop an efficient and accurate framework by performing force alignment on the 1-best ASR hypothesis. In particular, we propose to use binarized states sequences for alignment, which yields an EOS estimation robust to ASR hypothesis, and the estimation error is reduced by 28% compared to aligning on phoneme states. In addition, we further observe a 30% error reduction by applying the intermediate-stage embeddings of the encoder as additional features to compute the binary probabilities.

Index Terms— Endpoint detection, force alignment, Viterbi algorithm, speech recognition

1. INTRODUCTION

Recent years have witnessed an increasing prevalence of smart home devices with voice assistants such as Amazon Alexa and Google Home. These systems use an endpoint (EP) detector (see e.g. [1, 2, 3, 4, 5, 6]) to determine when the user has finished speaking automatically. Responsive EP detection is crucial to the user experience [7, 8], and therefore, a tremendous amount of effort has been spent in monitoring the EP latency of voice assistants. Given an utterance, the EP latency is defined as

$$\text{EP latency} := t_{\text{EP}} - t_{\text{EOS}}. \tag{1}$$

which is the difference between the time the EP is estimated, denoted by t_{EP}, and the time point of the true end-of-speech (EOS), denoted by t_{EOS}. Visually, the EOS is the point in time when the waveform of the user's utterance has fully decayed, as illustrated in the top plot of Fig. 1.

A technical problem for computing Eq. (1) is that the EOS cannot be directly obtained but needs to be estimated from the raw audio. Notably, having an accurate EOS estimation is challenging because: (1) in a typical case when background noise is present, the actual EOS timestamp could be buried and is therefore hard to be detected (see the bottom plot of Fig. 1); (2) given the large volume of incoming traffic, the estimation process must be operated in streaming fashion with a low computational cost due to the hardware constraints.

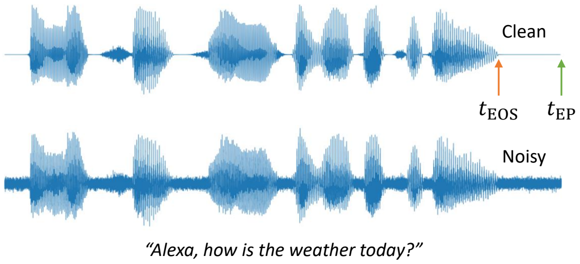

Fig. 1: *Top*: illustration of the endpoint t_{EP} and the end-of-speech t_{EOS}, on the utterance "*Alexa, how is the weather today*". *Bottom*: an example when background noise makes the true EOS timestamp hard to detect.

A conventional approach to EOS estimation is first to acquire the human transcript of the utterance, then performs a *force alignment* [9] based on a pre-trained hidden Markov model (HMM) that matches each audio frame to the phonemes of the transcript. In this way, the EOS is simply determined as the last audio frame that is labeled as a speech phone. Although this method is fairly accurate in general, it is impractical to be deployed in an online system due to its heavy computational complexity and the requirement of human transcription.

To address the limitations above, in this work, we propose an efficient and accurate framework for EOS estimation. Our system follows the same spirit of the force alignment-based algorithm mentioned above. However, it replaces the human transcript with the 1-best hypothesis output from an end-to-end ASR model such as the recurrent neural network transducer (RNN-T) [10] or Listen, Attend and Spell (LAS) [11]. Most importantly, to handle the error introduced by a wrong ASR hypothesis, we propose using binarized state sequences (instead of the phonemes) for alignment, which is robust to

[*] The author performed the work as an intern at Amazon Alexa.

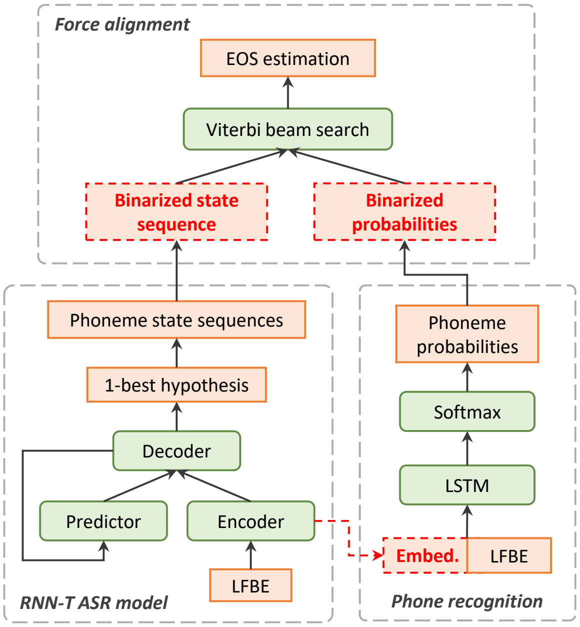

Fig. 2: An overview of our framework (and the baseline system) with three components: (1) the RNN-T ASR model; (2) the phone recognition network; (3) the force alignment step. Our modifications to the baseline system are indicated with red text and red dashed lines.

ASR hypothesis. Remarkably, this idea does not sacrifice any estimation accuracy but dramatically reduces the error compared to the traditional phoneme alignment. Besides, the HMM part in the original method is substituted by a small but effective long short-term memory (LSTM)-based phoneme recognition network. Here, by using embeddings of the RNN-T encoder as additional features to the network, we obtain a more accurate EOS estimation without introducing extra cost and latency. An overview of our framework is shown in Fig. 2.

The rest of this paper is organized as follows: we describe the baseline and our methods in Section 2. The corresponding experimental results are presented in Section 3. We end the paper with conclusions in Section 4.

2. PROPOSED EOS ESTIMATION MODEL

2.1. Baseline system

We first introduce the baseline EOS estimation system, which consists of an RNN-T ASR model, a phone recognition network, and a force alignment step. The phone recognition network is a 1-layer LSTM that maps the log-mel filterbank energies (LFBE) to senone probabilities, and is trained by using cross-entropy loss. Meanwhile, the RNN-T model generates its 1-best hypothesis, which is then converted to a senone state sequence by Kaldi [12]. The force alignment step uses Viterbi beam search [13] with a beam width of 8 to find the most likely alignment of the senone sequence given the frame-level senone probabilities. The EOS is then determined by finding the last senone state corresponding to speech. The baseline system is depicted in Fig. 2 (excluding the modules with the red dashed line).

A primary consideration for the design of the baseline system was low latency, so that the EOS can be estimated for all incoming traffic without causing backlogs. Hence, the phone recognition network is just a 1-layer LSTM, although stacking more layers improves the recognition performance, as shown in Table 1. Also, we remark that the phone recognition network is trained separately from the RNN-T. An advantage of this strategy is that either model can be updated independently. However, given the similarity of the ASR and phone recognition tasks, there exists room for improvement when the two models are coupled during training.

2.2. Proposed system

The baseline system suffers from two major problems. First, given the force alignment result essentially depends on the accuracy of phone recognition, using only 1-layer LSTM in the phone recognition network usually performs poorly. Second, given the senone states, the alignment result becomes unreliable when a wrong ASR hypothesis is provided. Therefore, to resolve the issues above, we propose two corresponding solutions described below.

2.2.1. Using RNN-T embeddings as additional features

In order to obtain a better phone recognition performance without introducing extra latency, we incorporate some existing intermediate embeddings from the RNN-T model as additional features to our phone recognition network. Here, we consider the output embeddings from the second layer of the RNN-T encoder to capture additional low-level acoustic features, which are useful for the phone recognition task. For each audio frame, the embeddings, denoted by $\boldsymbol{x}_1 \in \mathbb{R}^{d_1}$, are concatenated with the corresponding LFBE features $\boldsymbol{x}_2 \in \mathbb{R}^{d_1}$ to form the input features to the phone recognition network $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2] \in \mathbb{R}^{d_1+d_2}$ (see Fig. 3 for an illustration). For training the phone recognition network with the embeddings, the trained RNN-T model weights are frozen so that ASR performance isn't impacted. As we shall show in Table 3, incorporating these embeddings yields a significant boost in phone recognition and further the EOS estimation accuracy.

2.2.2. Force alignment on binarized states

As we mentioned earlier, force alignment on phoneme states could be inaccurate due to a wrong ASR hypothesis. Moreover, from the perspective of EOS estimation, aligning on the internal phoneme states within each word in the utterance is unnecessary, as EOS is only determined by the last speech

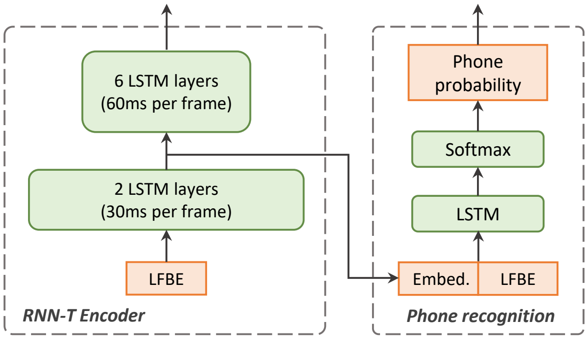

Fig. 3: Illustration of the embedding sharing between the RNN-T encoder and the phone recognition network. The output embeddings from the first two LSTM layers of the encoder is concatenated with the LFBE features and input into the phone recognition network.

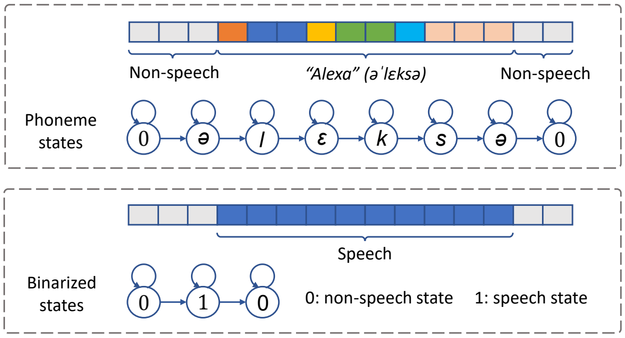

Fig. 4: Illustration of the phoneme and binarized states for the word *Alexa*. The sequence of phoneme states presented in the *top*, where the phonemes are indicated by different colors. In contrast, binarized states merge all the speech phonemes into one state indicated by '1', as shown in the *bottom*.

phone. These facts above motivate us to propose using a *binarized* state sequence instead. The idea is to treat all phonemes as a single state, indicated by one, and a non-speech state, indicated by zero. For instance, an utterance with only one word yields a sequence $0 \rightarrow 1 \rightarrow 0$. Fig. 4 provides an illustration. In this way, alignment on the binarized states becomes robust to ASR hypotheses as it does not rely on the specific words recognized by the ASR model, and no unnecessary effort is spent on aligning the internal speech phone states. Moreover, the computational cost for alignment (linear with the sequence length) is reduced due to a shorter binary state sequence.

Given the phoneme state sequence, the binarized sequence can be obtained by merging all the neighboring speech phone states to only one. To obtain the corresponding binarized probabilities needed for alignment, we sum over all the probabilities of speech phones output from the phone recognition model as the probability of 1 in the binarized states, and keep the non-speech probability unchanged (see Fig. 5).

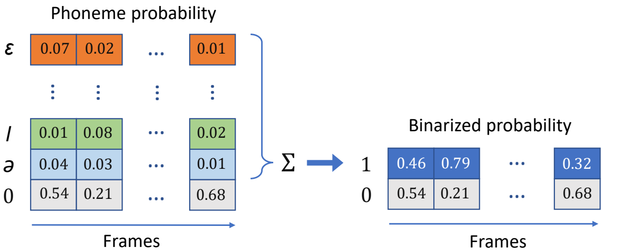

Fig. 5: Illustration of the binarized state probabilities. Given the phoneme probabilities, we keep the probability for the non-speech state (indicated by '0') unchanged, and sum over the probabilities for all the speech phone states into one category indicated by '1'.

In terms of the EOS estimation performance, we highlight that even in the case when the hypothesis is perfect, using binarized states does not sacrifice any accuracy but dramatically reduces the error compared to alignment on phonemes, as shown in Table 1.

3. EXPERIMENTS

3.1. Experiment setup

The baseline phone recognition model uses 64-dimensional log-Mel filter bank energy (LFBE) features with a frame rate of 1 frame every 30 ms. The phone recognition network consists of an LSTM layer with 1024 units, followed by a softmax layer to output phone probabilities. The RNN-T encoder consists of 8 layers, where the first 2 layers operate at the input frame rate, followed by 6 time-reduced layers. The embeddings from the second layer is of 1024 dimension.

3.2. Dataset

The RNN-T model and the phone recognition network are trained using far-field audio collected a single microphone channel. The data has an average SNR of 24 dB. The training set and test sets for the phone recognition network includes roughly 10^4 hours and 50 hours respectively of English utterances. The ground truth senone labels are generated by a pre-trained HMM with human transcription provided. All the utterances are de-identified.

3.3. Evaluation metrics

For each utterance, given the estimated EOS denoted $\hat{t}_{\text{EOS}}$ and the corresponding true EOS timestamp t_{EOS}, the error of EOS estimation is measured by

$$e_{\text{EOS}} := |\hat{t}_{\text{EOS}} - t_{\text{EOS}}|.$$

The true EOS timestamp is obtained from an offline forced alignment model that is accurate in detecting the speech boundaries but is computationally slow.

Table 1: EOS estimation error by using phoneme states and binarized states with different number of LSTM layers. We report the relative percentage change compared with the baseline (negative percentage change is better).

Layers	Phoneme states			Binarized states		
	mean	std	DTM	mean	std	DTM
1	Baseline system			-15.6	+3.4	-28.7
2	-10.4	-13.2	-13.9	-21.1	-16.2	-38.9
3	-18.7	-19.6	-16.8	-28.3	-19.9	-40.3
4	-27.1	-26.3	-29.6	-35.5	-29.1	-47.0
5	-27.1	-24.1	-29.6	-34.9	-20.7	-45.4

We summarize the EOS estimation error on the test set with the following three metrics: (1) mean, (2) standard deviation (std), and (3) double-trimmed mean (DTM), which is the average of the errors between the 95th and 99th percentiles. It serves as a measure of tail performance.

3.4. Results

We first evaluate the idea of binarized states given in Section 2.2.2. Note that the RNN-T embeddings are not included as additional features to the phone recognition network in these results. In order to better demonstrate improvement on the EOS estimation result by using the binarized states (compared to the phoneme states), the force alignment is performed on the human transcript (instead of the ASR hypothesis). The corresponding EOS estimation result is given in Table 5. Here, the phone recognition network is trained with a different number of LSTM layers. All the metrics are reported as the relative improvement or degradation compared to the Baseline model (with phoneme states and 1 LSTM layer). As expected, we first observe the EOS estimation error decreases as the number of layers increases. More importantly, given the same number of layers, using binarized states generally yields over 10% and 20% reduction on the mean and DTM of the EOS estimation error, demonstrating the efficacy of binarized states, especially on those extreme cases with a high estimation error.

Recall that in Section 2.2.2, our binarized probabilities comes by summing over all speech phone probabilities, as shown in Fig. 5. Alternatively, the probabilities can be obtained directly by training a binary classification model where the output dimension is reduced to two, which indicates speech and non-speech. This idea, intuitively, is more straightforward and requires less computational complexity. However, as shown in Table 2, training such a binarized recognition model even yields a worse EOS estimation accuracy than the baseline, which suggests the necessity of our idea of computing the binarized probabilities.

Now we move to the method of using RNN-T embeddings as additional features. The corresponding EOS estimation result is given in Table 3. Here, the phone recognition model consists of only one LSTM layer. As we can see, by incorporating the RNN-T embeddings, we obtain over 26% reduction on the mean and variance of the EOS estimation error, compared to the baseline model, and 35% on the DTM 95:99. Moreover, by further applying the idea of binarized states, we observe over 30% and 44% reduction on the mean and DTM 95:99 of the error. These results demonstrate the importance of RNN-T embeddings to the EOS estimation accuracy.

Table 2: EOS estimation error when training a binary speech/non-speech classification model directly instead of binarizing the phone probabilities. We report the relative percentage change compared with the baseline (negative percentage change is better).

Model	mean	std	DTM
Binary classification model	+16.3	+8.1	+3.7
Phone recognition model + Binarized probabilities	-15.6	+3.4	-28.7

Table 3: EOS estimation error by using RNN-T embeddings as additional features (and the binarized states). We report the relative improvement (degradation) in percentage compared to the baseline model.

Model	mean	std	DTM
Use RNN-T embedding	-26.5	-26.9	-34.4
Use RNN-T embedding + Binarized states	-30.7	-26.6	-44.5

4. CONCLUSIONS

In this work, we introduce an end-of-speech estimation problem, which arises when measuring the endpoint detection latency. A corresponding framework is proposed for estimating EOS accurately with low latency that can be deployed for online streaming audio applications. In particular, two methods (using the RNN-T embeddings and the binarized states) are designed to increase the EOS estimation accuracy and its robustness to the ASR hypothesis. Experimental result shows the improvement of our model compared to the baseline system. To improve upon our work, we will explore incorporating a small LSTM to refine the EOS estimation. We will also explore using embeddings from the RNN-T decoder to include lexical information in EOS estimation.

5. REFERENCES

[1] Bo Li, Shuo-yiin Chang, Tara N Sainath, Ruoming Pang, Yanzhang He, Trevor Strohman, and Yonghui Wu, “Towards fast and accurate streaming end-to-end asr,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6069–6073.

[2] Shuo-Yiin Chang, Rohit Prabhavalkar, Yanzhang He, Tara N Sainath, and Gabor Simko, “Joint endpointing and decoding with end-to-end models,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5626–5630.

[3] Roland Maas, Ariya Rastrow, Chengyuan Ma, Guitang Lan, Kyle Goehner, Gautam Tiwari, Shaun Joseph, and Björn Hoffmeister, “Combining acoustic embeddings and decoding features for end-of-utterance detection in real-time far-field speech recognition systems,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5544–5548.

[4] Shuo-Yiin Chang, Bo Li, Tara N Sainath, Gabor Simko, and Carolina Parada, “Endpoint detection using grid long short-term memory networks for streaming speech recognition.,” in *Interspeech*, 2017, pp. 3812–3816.

[5] Tara N Sainath, Ruoming Pang, David Rybach, Basi García, and Trevor Strohman, “Emitting word timings with end-to-end models.,” in *INTERSPEECH*, 2020, pp. 3615–3619.

[6] Liang Lu, Jinyu Li, and Yifan Gong, “Endpoint detection for streaming end-to-end multi-talker asr,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7312–7316.

[7] Nigel G Ward, Anais G Rivera, Karen Ward, and David G Novick, “Root causes of lost time and user stress in a simple dialog system,” 2005.

[8] A Raux, D Bohus, B Langner, AW Black, and M Eskenazi, “Doing research on a deployed spoken dialogue system: one year of lets go! experience research on a deployed spoken dialogue system: one year of lets go! experience. proceedings of interspeech,” 2006.

[9] Daniel Jurafsky and James H Martin, “Speech and language processing: An introduction to speech recognition, computational linguistics and natural language processing,” *Upper Saddle River, NJ: Prentice Hall*, 2008.

[10] Alex Graves and Navdeep Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in *International conference on machine learning*. PMLR, 2014, pp. 1764–1772.

[11] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[12] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., “The kaldi speech recognition toolkit,” in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number CONF.

[13] Andrew Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.