INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Utterance Confidence Measure for End-to-End Speech Recognition with Applications to Distributed Speech Recognition Scenarios

Ankur Kumar, Sachin Singh*, Dhananjaya Gowda*,*
Abhinav Garg, Shatrughan Singh, Chanwoo Kim

Samsung Research

{ankur.k, singh.sachin, d.gowda, abhinav.garg, shatrughan.s, chanw.com}@samsung.com

Abstract

In this paper, we present techniques to compute confidence score on the predictions made by an end-to-end speech recognition model. Our proposed neural confidence measure (NCM) is trained as a binary classification task to accept or reject an end-to-end speech recognition result. We incorporate features from an encoder, a decoder, and an attention block of the attention-based end-to-end speech recognition model to improve NCM significantly. We observe that using information from multiple beams further improves the performance. As a case study of this NCM, we consider an application of the utterance-level confidence score in a distributed speech recognition environment with two or more speech recognition systems running on different platforms with varying resource capabilities. We show that around 57% computation on a resource-rich high-end platform (*e.g.* a cloud platform) can be saved without sacrificing accuracy compared to the high-end only solution. Around 70-80% of computations can be saved if we allow a degradation of word error rates to within 5-10% relative to the high-end solution.
Index Terms: confidence measure, end-to-end speech recognition, attention models, distributed speech recognition

1. Introduction

Recent works have focused on streaming end-to-end (e2e) offline ASR systems capable of running on resource-constrained platforms with reduced latency [1, 2, 3]. The performance of these streaming architectures have improved significantly on large scale corpora and are comparable in accuracy to the much bigger speech recognition models that run on resource rich platforms. However, they still lag behind these high-end speech recognition models because of limited computational resources, which mandates the need for model compression, quantization, and other optimizations [1, 4, 5, 3, 6, 7, 8]. One alternative is to adopt a distributed ASR solution with at least two ASR engines, one low-end small-footprint ASR engine running on a resource-constrained platform, and another high-end ASR running on a resource-rich platform. In such scenarios, we can first route the incoming traffic to the low-end engine, and if the quality of its prediction is poor, the audio stream can be directed to the high-end engine.

The above objective can be achieved if we have access to a measure of confidence on the predicted output. Based on the confidence score for the low-end ASR hypothesis and a threshold value, an accept/reject decision can be made. The hypothesis should be accepted when it is at least as accurate as the high-end ASR prediction; otherwise, it should be rejected, and the query is handled by the high-end ASR. It is imminent that any low-end ASR prediction accepted results in computation savings for the high-end ASR. This motivates us to explore utterance-level confidence measures, especially for low-end speech recognition models.

Many approaches have been explored for word confidence measures in the context of conventional speech recognition systems [9]. We can broadly classify them into three categories a) methods based on posterior probabilities, b) framework of utterance verification, and c) techniques using a combination of predictor features. The first approach is non-parametric and requires post-processing of posterior probabilities to obtain a confidence score. Calculating accurate word posterior probability is challenging [10], and therefore, techniques belonging to this category investigate approximate methods to obtain the probability from a word graph or n-best hypothesis [11, 12, 13, 14]. The second category of utterance verification technique uses a framework of statistical hypothesis testing to train two complementary models, one for *null* hypothesis and another for alternative hypothesis. Likelihood ratio testing is used to test the two models in order to arrive at an accept/reject decision for the recognition result. The major challenge in this approach is to model the complex alternate hypothesis accurately [15, 16].

The last class of techniques is based on predictor features. A binary classifier is trained to classify a word as correct or incorrect based on a combination of features from speech recognition model, which are called predictor features. The posterior probability of a word being correct given predictor features is used as a measure of confidence. Most of the works in this direction focus on deriving increasingly discriminating set of predictor features for the task [17, 18, 19, 20, 21, 12]. Another aspect of the predictor feature-based methods is the choice of binary classifier. There is an increasing trend to use neural network-based classifiers [22, 23, 19], which clearly outperforms non-neural based classifiers.

To the best of our knowledge, this work is the first to explore utterance confidence measures in the context of e2e ASR. We investigate two approaches for utterance-level confidence score in this paper: a) sequence posterior probability-based method and b) predictor feature-based technique. For the first approach, we aggregate label-level confidence scores as proposed in [24] to obtain a confidence score for the entire hypothesis. However, experiments show that this score performs poorly. We propose a simple extension of this technique, which brings significant performance gains over the baseline method. For the predictor feature-based method, a neural network-based binary classifier is trained with features derived from e2e ASR. We consider encoder output, decoder output, attention weights, and beam scores in the experiments. Our work is the first to utilize attention weights as well as decoder output and attention weights from multiple beams for ASR confidence measure. Experimental results show the importance of each of these fea-

*Equal contribution.

http://dx.doi.org/10.21437/Interspeech.2020-3216

tures and that it is possible to learn a neural confidence measure or model (NCM) from as small as ∼20 hours of speech data (20K utterances). Finally, a novel metric is proposed to evaluate the performance of utterance confidence measures as a tradeoff between the overall ASR accuracy and computations saved on a high-end platform in a distributed ASR scenario. Results show that the proposed metric is consistent with standard metrics used in the literature for confidence measures and gives a more meaningful evaluation of the confidence measure in a practical scenario.

2. Utterance confidence measures

2.1. Word density confidence measure

Let $\{h_i\}_{i=1}^n$ be the n-best hypotheses, and $\{s_i\}_{i=1}^n$ be the corresponding scores obtained by the speech recognition system. Word density confidence measure (WDCM) [24] assigns a confidence score to each token (word or a subword) present in the best hypothesis as follows:

Step 1: Convert scores $\{s_i\}$ to hypothesis probabilities after scaling with some suitable factor a:

$$p_{h_i} = \frac{\exp(-as_i)}{\sum_{k=1}^{n} \exp(-as_k)}, \tag{1}$$

Step 2: For each token w_k in the best hypothesis $h_1 = \{w_k\}_{k=1}^K$, compute the token confidence as:

$$p_{w_k} = \sum_{i=1}^{n} \delta(w_k, h_i) p_{h_i}, \tag{2}$$

$$\text{where} \quad \delta(w_k, h_i) = \begin{cases} 1 & \text{if} \quad w_k \in Align(h_1, h_i), \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

We can average these scores for each token to get an utterance-level confidence score for ASR output as follows:

$$C_{wd} = \frac{1}{K} \sum_{k=1}^{K} p_{w_k} \tag{4}$$

2.2. Beam-scatter weighted WDCM

WDCM score for an utterance, as described in the previous section, is often inflated. It helps if we scale the score according to *beam-scatter* or relative beam probabilities. We consider the top two beams (chosen empirically) for this purpose and define beam scatter weighted WDCM (BWDCM) score as:

$$C_{bwd} = \frac{1}{1 + e^{-\lambda(p_{h_1} - p_{h_2})}} C_{wd} \tag{5}$$

where p_{h_1} and p_{h_2} are top two beam probabilities respectively obtained in Step 1 of WDCM, and λ is a tunable parameter.

Our extension of the baseline WDCM approach is supported by observation in [24] that the scores of competing hypotheses are among the most useful features for confidence measure. The interpretation of beam scatter fits accordingly, which is as follows. If the top two beams have similar ASR scores, it means that ASR output is ambiguous, and we accordingly halve the WDCM score. A high ASR score for the top beam, compared to the rest of the beams, indicates that the speech recognition model is confident in its prediction. In this case, the BWDCM score is almost equal to the WDCM score, given that λ is sufficiently large.

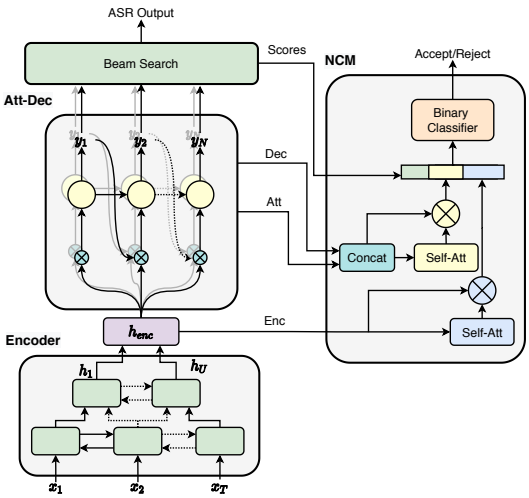

Figure 1: *Proposed NCM architecture with input features from e2e ASR. Scores, Enc, Dec and Att features shown here are beam scores, encoder output, decoder output and attention weights respectively, as described in Section 2.3*

2.3. Neural confidence measure

Typically, a well-trained speech recognition model assigns a high score to the top-most beam if it is *well-formed* and the model is confident, and relatively lower scores for other beams. BWDCM technique is based on this assumption and works well in such scenarios. However, in some cases, the speech recognition models can produce almost similar scores for all the beams. Also, speech recognition models are commonly biased to perform well in the development setup and can throw up different patterns in the production setup. This can lead to violations of some of the assumptions made in designing non-neural methods such as WDCM or BWDCM. In yet another case, we observed that *null* hypothesis frequently appeared as the second-best beam with a score comparable to the top-beam when our e2e speech recognition model was biased on short queries. All these reasons make the BWDCM method perform poorly on several utterances.

In view of this, we propose to use neural confidence measures or models (NCM), which do not make such assumptions and are trained with features derived from speech recognition model ready for deployment, as shown in Figure 1. We formulate the utterance-level confidence measure as a binary classification task. NCM learns to predict 1 if ASR output has no error, otherwise 0. It is based on the observation that a well-trained speech recognition model has a low sentence error rate. Therefore, learning to distinguish between ASR predictions with zero and non-zero errors should result in significant computation savings. We make use of the following features from e2e speech recognition model for NCM experiments:

Beam scores (Scores): These are log-probability scores assigned by e2e ASR decoder to each beam. Word density-based approaches accept or reject an utterance based on some post-processing of these scores. We generalize the approaches to some extent by using neural network with these scores as input.

Encoder output (Enc): It acts as a summary of the acoustic input to the speech recognition model. [21] shows that acoustic embedding is a useful feature for confidence score estimation. We expect decoder output, described next, to capture this information. However, experiments show that using both the features

results in significant gains over NCM using any one of the two features.

Decoder output (Dec): We accumulate decoder output logits (top-K only) for each token in the top-most beam. This is because only these scores affect beam search decoding. It also helps in reducing the number of neural network parameters as ASR output vocabulary size is very large.

Attention weights (Att): Attention patterns may reflect the quality of ASR output. Attention weights in a decoding step are typically concentrated over a small number of encoder timesteps. Therefore, a fuzzy attention distribution may indicate the possibility of an error in the output.

Multi-beam NCM: Different from previous works, we incorporate decoder output and attention weight features discussed above for the rest of the beams also. We have seen in Section 2.2 that competing hypotheses are very useful source of information. Therefore, we expect multi-beam NCM to outperform NCM using features from top-beam only.

3. Case study: Distributed ASR scenario

In this section, as an application of this NCM, we consider a new scenario where we use utterance confidence measures for a distributed ASR system consisting of low-end and high-end speech recognition models. The objective is to reduce the overall computational cost on the high-end ASR without sacrificing the performance significantly.

3.1. Distributed speech recognition scenario

In this case study, we use a streaming encoder-decoder architecture with Monotonic Chunkwise Attention (MoChA) [1, 25] as the low-end speech recognition model. The encoder consists of 6 unidirectional Long Short-Term Memory (ULSTM) [26] layers with an overall temporal sub-sampling factor of 8. The high-end speech recognition model shares the encoder of the low-end model, but adds a backward LSTM layer to the top-most forward LSTM layer of the shared encoder and uses a full-attention decoder [27, 28, 29, 30]. Our motivation for using this kind of shared encoder architecture is that it gives us an option of transmitting only the shared encoder embeddings to the high-end ASR, as a future work [29]. All the LSTM layers have 1536 hidden units. Decoder, for both cases, is an LSTM cell with 1000 hidden units followed by maxout and softmax layers. For both speech recognition systems, we use the power-mel feature [31, 6] with a power coefficient of 1/15. Output vocabulary contains around 10K BPE subword units. Both the models are trained jointly [29] on $\sim$10K hours of English internal speech corpus described in [1]. The models are trained for around 14 full epochs using Adam optimizer. The training system was built *in-house* using the Tensorflow 2 Keras APIs from scratch. The low-end speech recognition system uses a beam size of 4, whereas the high-end one has 12 beams during the beam search. Model performances are evaluated on a test set with 1585 utterances. The low-end and the high-end speech recognition models have word error rates (WER) of 14.57% and 10.40%, respectively, on this test set. The distributed speech recognition scenario outlined in this paper is only an experimental setup to give a more meaningful interpretation to the evaluation of different confidence measures. This solution has not been evaluated for latency or other requirements of a real-world production setup.

3.2. Conventional evaluation of confidence measures

Conventionally, the performance of a confidence measure is evaluated using the receiver operating characteristics (ROC) curve which shows the tradeoff between *true positive rate* and *false positive rate*. Area under the ROC curve (AUC) and equal error rate (EER), the rate where *false positive* and *false negative* rates are equal, are popularly used. Normalized Cross Entropy (NCE) is also widely used [9]. A better confidence measure should have higher value of AUC as well as NCE and lower value of EER. These metrics are good to compare any two methods but do not convey any practical meaning to the numbers.

3.3. Computation saved *vs* relative increase in error rate

In a distributed ASR scenario, the low-end ASR prediction is either accepted or rejected based on the confidence score and a threshold value. If it is rejected, the audio is transmitted to the high-end ASR to get a new prediction. In such a scenario, we define the computation saved (CS) as a fraction of total utterances that were classified as accepted on the low-end device or platform. A combined word error rate (WER) is calculated for the distributed scenario, where low-end ASR prediction is used for accepted utterances and the high-end ASR prediction is used for rejected utterances. Relative increase in error rate (RIER) is defined as the relative difference between WERs of the distributed ASR and the high-end ASR. When all utterances are accepted, the combined WER approaches that of the low-end ASR with 100% CS, and on the other extreme, it approaches that of the high-end ASR with 0% CS when all utterances are rejected.

4. Experiments and results

4.1. Neural confidence measure

NCM is a binary classifier with two feed-forward layers, each having 64 hidden units and RELU non-linearity. Inputs to the network are features derived from the low-end speech recognition model. Table 1 lists all the feature combinations used in our experiments, along with their performance on the task. Scores is a 4-dimensional feature vector containing all the beam scores. Decoder output for a beam is a sequence of 10-dimensional feature vector. Three techniques were tried to summarize such output into a fixed-length vector – averaging along temporal dimension, weighted average using self-attention [32], and LSTM followed by self-attention. Adding an LSTM layer to accumulate the information better did not have any significant improvement. Therefore, we use self-attention based weighted averaging in all our experiments, which was marginally better than simple averaging. Similarly, encoder output is also transformed into a fixed-length vector of 1536 dimensions. Attention weights is a sequence of 2-dimensional feature vector since we use a chunk size of 2 for MoChA. These weights are concatenated with the corresponding decoder output in feature dimension before summarizing. For multi-beam NCM, we summarize decoder (with attention) features for each beam separately and then concatenate. These fixed-length vectors (depending on feature combination) are concatenated and fed to the binary classifier to predict a 0/1 label.

4.2. Training data for NCM

NCM is trained on the devset (20K utterances) that was used as a validation set for the model training [21]. Input features are obtained by running the trained low-end speech recognition model with a beam search. If the prediction exactly matches the reference transcription, then the NCM target is set to one, otherwise zero. The positive class probability assigned by a trained NCM is used to make an accept/reject decision at the test time.

Table 1: *Comparison of utterance confidence measures using standard metrics (AUC, NCE and EER) as well as the proposed CS-vs-RIER for a distributed ASR scenario. 'ASR' method in first row refer to using the sequence probability scores assigned by e2e speech recognition models to the top-most beam as utterance-level confidence score. Input features are as described in Section 2.3*

	Input Features								CS @ x% RIER		
	Scores	Enc	Dec	Att	Beams	AUC ↑	NCE ↑	EER ↓	$x = 0$ ↑	$x = 5$ ↑	$x = 10$ ↑
ASR	–	–	–	–	–	**0.76**	**0.12**	**0.33**	0.0	0.20	0.59
WDCM	✓	–	–	–	–	0.61	-0.58	0.40	0.0	0.17	0.51
BWDCM	✓	–	–	–	–	0.73	0.11	**0.33**	0.0	**0.41**	**0.62**
NCM	✓	–	–	–	–	0.85	0.27	0.22	0.0	0.44	0.77
	✓	✓	–	–	–	0.88	0.32	0.21	0.0	0.66	0.81
	✓	–	✓	–	Top	0.89	0.36	0.19	0.11	0.69	0.77
	✓	✓	✓	–	Top	0.89	0.39	0.19	0.40	0.71	**0.83**
	✓	✓	✓	✓	Top	**0.90**	0.40	0.19	0.46	0.71	0.81
	✓	✓	✓	–	All	**0.90**	**0.41**	**0.18**	0.51	0.71	0.80
	✓	✓	✓	✓	All	**0.90**	**0.41**	0.19	**0.57**	**0.74**	0.81

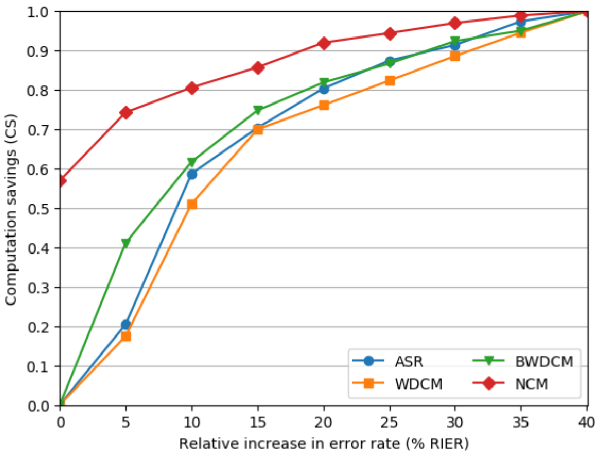

Figure 2: *CS vs RIER tradeoff for different confidence measures. ASR confidence measure uses probability assigned by e2e ASR to the top-most beam as confidence score*

4.3. Results

Table 1 lists results for all our experiments. ASR confidence measure, which uses probability assigned by (low-end) e2e ASR to the top-most beam, gives only 20% computation saving at 5% RIER. WDCM, a popular technique for token-level confidence measure, performs poorly when utilized for utterance-level decision. Even more surprising is the high negative NCE score of -0.58 for WDCM, which means that using prior class probabilities is a much better alternative. This also highlights the effect of beam scatter due to which BWDCM score performs much better than even ASR confidence measure, and is close to the baseline NCM model (only beam scores as input) performance with 41% CS at 5% RIER.

The CS-*vs*-RIER tradeoff curves for various thresholds are plotted in Figure 2 to evaluate different confidence measure methods. A better confidence score model should have a high value of CS at 0% RIER. From Figure 2, NCM performs the best with 57% computation saved at 0% RIER, whereas non-parametric methods have 0% computation saved.

NCM consistently outperforms all the other techniques, which is not surprising given it is a parametric model. A large gap in the performance for ASR confidence measure (first row in Table 1) and baseline NCM (fourth row) indicates that there is indeed a need for calibration of the sequence probability scores assigned by speech recognition model to the best hypothesis. Performance gain for models using additional ASR features is consistent with the importance of using a rich feature set for predictor feature-based confidence measures. NCM trained on beam scores and decoder output has higher scores on all the metrics than the corresponding experiment with encoder output. This suggests that decoder feature is much more discriminating than encoder embedding, which is what we had expected. Moreover, using both the features together perform even better with 40% server computation savings at 0% RIER. NCM performance on all the metrics except CS-*vs*-RIER tradeoff seems to saturate when sufficient features have been incorporated. However, our proposed metric is still able to distinguish between different versions of NCM models using features from top-beam and multi-beam, as well as with and without attention weights. According to NCM performances on standard metrics, using attention weights or deriving features from multiple beams does not seem to be effective, which is not true. Therefore, it is a better strategy to define a custom metric that is consistent with the standard metrics and closely captures the objective. Finally, NCM, with all the features, perform the best with 57% computation saved at 0% RIER, which goes up to 74% with a small degradation in the combined distributed ASR transcription quality. This essentially means that low-end speech recognition model performance for the accepted utterances matches the high-end speech recognition model performance, which can be a simple alternative to complex techniques required to improve ASR performance on resource-constrained low-end platforms.

5. Conclusions

In this paper, we present a new approach to compute the confidence score on the predictions made by an end-to-end speech recognition model. We incorporate features from an encoder, a decoder, and an attention block of the attention-based end-to-end speech recognition model in this NCM model. In the experimental results, it has been shown that this new NCM model significantly outperforms the conventional WDCM and BWDCM approaches. In a case study of using this NCM model, the proposed method results in more than 70% computation saved in a distributed speech recognition scenario without significantly compromising the performance.

6. References

[1] K. Kim, K. Lee, D. Gowda, J. Park, S. Kim, S. Jin, Y.-Y. Lee, J. Yeo, D. Kim, S. Jung, J. Lee, M. Han, and C. Kim, "Attention based on-device streaming speech recognition with large speech corpus," in *Proc. ASRU*, Dec. 2019, pp. 956–963.

[2] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *Proc. ICASSP*. IEEE, 2019, pp. 6381–6385.

[3] A. Garg, G. Vadisetti, D. Gowda, S. Jin, A. Jayasimha, Y. Han, J. Kim, J. Park, K. Kim, S. Kim, Y. Lee, K. Min, and C. Kim, "Streaming on-device end-to-end asr system for privacy-sensitive voicetyping," in *Proc. Interspeech*, 2020.

[4] A. Garg, D. Gowda, A. Kumar, K. Kim, M. Kumar, and C. Kim, "Improved multi-stage training of online attention-based encoder-decoder models," in *Proc. ASRU*. IEEE, 2019, pp. 70–77.

[5] D. Lee, P. Kapoor, and B. Kim, "Deeptwist: Learning model compression via occasional weight distortion," *CoRR*, vol. abs/1810.12823, 2018. [Online]. Available: http://arxiv.org/abs/1810.12823

[6] C. Kim, M. Kumar, K. Kim, and D. Gowda, "Power-law nonlinearity with maximally uniform distribution criterion for improved neural network training in automatic speech recognition," in *Proc. ASRU*, Dec. 2019, pp. 988–995.

[7] A. Garg, A. Gupta, D. Gowda, S. Singh, and C. Kim, "Hierarchical multi-stage word-to-grapheme named entity corrector for automatic speech recognition," in *Proc. Interspeech*, 2020.

[8] C. Kim, S. Kim, K. Kim, M. Kumar, J. Kim, K. Lee, C. Han, A. Garg, E. Kim, M. Shin, S. Singh, L. Heck, and D. Gowda, "End-to-end training of a large vocabulary end-to-end speech recognition system," in *Proc. ASRU*, 2019.

[9] H. Jiang, "Confidence measures for speech recognition: A survey," *Speech communication*, vol. 45, no. 4, pp. 455–470, 2005.

[10] F. Wessel, R. Schluter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Transactions on speech and audio processing*, vol. 9, no. 3, pp. 288–298, 2001.

[11] F. Wessel, K. Macherey, and H. Ney, "A comparison of word graph and n-best list based confidence measures," in *EUROSPEECH*, 1999.

[12] T. Kemp and T. Schaaf, "Estimating confidence using word lattices," in *Fifth European Conference on Speech Communication and Technology*, 1997.

[13] F. Wessel, K. Macherey, and R. Schluter, "Using word probabilities as confidence measures," in *Proc. ICASSP*, vol. 1. IEEE, 1998, pp. 225–228.

[14] S. O. Kamppari and T. J. Hazen, "Word and phone level acoustic confidence scoring," in *Proc. ICASSP*, vol. 3, 2000, pp. 1799–1802 vol.3.

[15] M. G. Rahim, Chin-Hui Lee, and Biing-Hwang Juang, "Discriminative utterance verification for connected digits recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 3, pp. 266–277, 1997.

[16] R. A. Sukkar and C.-H. Lee, "Vocabulary independent discriminative utterance verification for nonkeyword rejection in subword based speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 6, pp. 420–429, 1996.

[17] R. San-Segundo, B. Pellom, K. Hacioglu, W. Ward, and J. M. Pardo, "Confidence measures for spoken dialogue systems," in *Proc. ICASSP*, vol. 1. IEEE, 2001, pp. 393–396.

[18] M. Benıtez, A. Rubio, P. Garcıa, and A. de la Torre, "Different confidence measures for word verification in speech recognition," *Speech Communication*, vol. 32, no. 1-2, pp. 79–94, 2000.

[19] M. A. Del-Agua, A. Gimenez, A. Sanchis, J. Civera, and A. Juan, "Speaker-adapted confidence measures for ASR using deep bidirectional recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 7, pp. 1198–1206, 2018.

[20] P.-S. Huang, K. Kumar, C. Liu, Y. Gong, and L. Deng, "Predicting speech recognition confidence using deep learning with word identity and score features," in *Proc. ICASSP*. IEEE, 2013, pp. 7413–7417.

[21] P. Swarup, R. Maas, S. Garimella, S. H. Mallidi, and B. Hoffmeister, "Improving ASR confidence scores for Alexa using acoustic and hypothesis embeddings," in *Proc. Interspeech*, vol. 2019, 2019, pp. 2175–2179.

[22] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, and A. Stolcke, "Neural-network based measures of confidence for word recognition," in *Proc. ICASSP*, vol. 2. IEEE, 1997, pp. 887–890.

[23] K. Kalgaonkar, C. Liu, Y. Gong, and K. Yao, "Estimating confidence scores on ASR results using recurrent neural networks," in *Proc. ICASSP*. IEEE, 2015, pp. 4999–5003.

[24] B. Rueber, "Obtaining confidence measures from sentence probabilities," in *Fifth European Conference on Speech Communication and Technology*, 1997.

[25] C. Chiu and C. Raffel, "Monotonic chunkwise attention," *CoRR*, vol. abs/1712.05382, 2017. [Online]. Available: http://arxiv.org/abs/1712.05382

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, no. 9, pp. 1735–1780, Nov. 1997.

[27] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. NIPS*, 2015, pp. 577–585.

[28] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*. IEEE, 2016, pp. 4960–4964.

[29] D. Gowda, A. Kumar, K. Kim, H. Yang, A. Garg, S. Singh, J. Kim, M. Kumar, S. Jin, S. Singh, and C. Kim, "Utterance invariant training for hybrid two-pass end-to-end speech recognition," in *Proc. Interspeech*, 2020.

[30] D. Gowda, A. Garg, K. Kim, M. Kumar, and C. Kim, "Multi-task multi-resolution char-to-bpe cross-attention decoder for end-to-end speech recognition," in *Proc. Interspeech*, 2019.

[31] C. Kim, M. Shin, A. Garg, and D. Gowda, "Improved vocal tract length perturbation for a state-of-the-art end-to-end speech recognition system," in *Proc. Interspeech*, 2019, pp. 739–743.

[32] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130*, 2017.