INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Word-level Confidence Estimation for CTC Models

Burin Naowarat, Thananchai Kongthaworn, Ekapol Chuangsuwanich

Chulalongkorn University, Bangkok, Thailand

6270145221@student.chula.ac.th, thananchai.ktw@gmail.com, ekapol.c@chula.ac.th

Abstract

Measuring confidence in Automatic Speech Recognition (ASR) is important for ensuring the reliability of downstream applications. Previous works proposed Confidence Estimation Module (CEM) for predicting confidences for autoregressive attention-based and neural transducer architectures. However, CEM for connectionist temporal classification (CTC) models have not been explored. In this work, we expand the idea of CEM to CTC models and further propose considering surrounding words for estimating confidences. Our experiments on four test sets in two languages demonstrate that our proposed method significantly reduces calibration errors of both common and rare words compared to naive confidences from CTC softmax. Moreover, we show that the approach is also effective for hard words and out-of-domain test sets, indicating its potential to be used as a reliable trigger for human intervention.

Index Terms: speech recognition, confidence estimation, CTC models

1. Introduction

Confidence scores play a vital role in ensuring the reliability and trustworthiness of various applications since the scores could tell how much the users can trust the systems [1, 2, 3]. In semi-supervised learning, confidence scores are used to manage the quality of pseudo-labeled data, preventing noisy transcriptions from harming the model's performance [4, 5, 6, 7]. In autonomous systems, confidence scores serve as a crucial fail-safe mechanism. They can be utilized to select between models that perform well at different characteristics of levels of data complexity [8] or notify human operators to intervene when the scores fall below a predetermined threshold [1, 2]. In the case of Automatic Speech Recognition (ASR), confidence scores can be easily measured in traditional hybrid ASR frameworks [9]. However, obtaining reliable scores for recent end-to-end ASR models is not straightforward [10].

End-to-end neural networks have become increasingly popular in ASR due to their simplified data pipeline and superior performance compared to hybrid systems [11, 12, 13]. These models produce probabilities for each output unit in the predefined set of tokens directly using a softmax function, allowing for straightforward use of these probabilities as confidence measures [14]. However, recent research has shown that these probabilities are often unreliable as the models tend to be overconfident in their predictions [15]. This overconfidence can make it difficult to accurately assess the reliability of ASR outputs. To address this issue, researchers have developed methods to calibrate the softmax probabilities [15] and/or estimate the confidence using an external predictive model such as Confidence Estimation Module (CEM) [8, 16, 17, 18].

CEM utilizes relevant features extracted from an ASR model to estimate confidence scores for predicted tokens or words. However, previous works have mainly focused on building CEM for attention-based sequence-to-sequence [16] and neural transducer networks [8, 17, 18, 19], which are resource-intensive models. On the other hand, the effectiveness of CEM for a more resource-friendly ASR model, connectionist temporal classification (CTC), remains underexplored. Despite its lower performance compared to the first two models, CTC-based ASR requires less computational power and memory footprints during training [20, 21, 22, 23]. Additionally, CTC is still dominantly used in other relative sequence-to-sequence tasks such as handwriting recognition [24, 25, 26]. Therefore, in this work, we aim to extend the use of CEM to CTC models.

In this paper, we propose a novel method for estimating word confidences of CTC predictions using a CEM. We introduce new features for the CEM and develop an algorithm that extracts these word-level features from the CTC softmax outputs. We evaluate a simple multi-layer perceptron (MLP) CEM that independently predicts confidence for each word, and propose to use Transformer CEM in order to take the entire sentence into account. Unlike [16], our CEM directly produces word confidences from word-level features. To the best of our knowledge, this is the first paper that uses CEM for CTC models and evaluates the impact of considering the entire sequence of CTC predictions for confidence estimation.

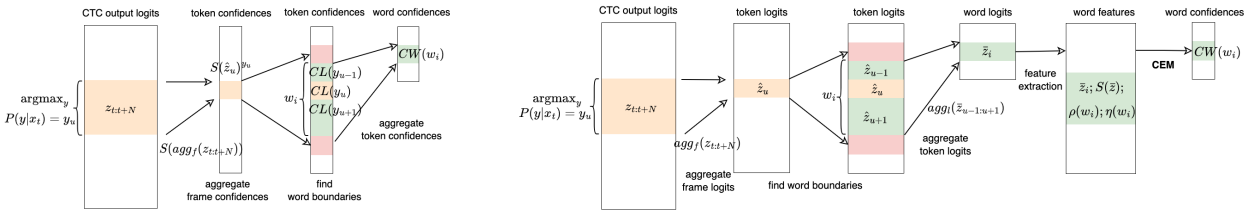
Our experimental results demonstrate that both variations of the proposed CEM significantly outperform naive confidences derived from CTC softmax on four test sets in two different languages, Thai and English. We observe substantial improvements in alignments between confidences and correction rates on LibriSpeech test-clean/test-other [27] and CommonVoice [28], as well as on Thai Podcast [29] and mock technical interviews test sets. Moreover, we show that the Transformer CEM, which considers the surrounding context, is more robust to unseen domains compared to the other methodologies we tested.

2. CTC Model

CTC model [30] is trained to maximize probabilities of predicting a transcription $y = (y_1, y_2, ..., y_U)$ for an input sequence $x = (x_1, x_2, ..., x_T)$. CTC model does not require ground-truth alignments for training, the model increases the likelihood of any possible alignments that can represent the target transcription, as shown in (1).

$$P(y|x) = \sum_{\pi:\mathcal{B}(\pi)=y} P(\pi|x) = \sum_{\pi:\mathcal{B}(\pi)=y} \prod_{t=1}^{T} P(\pi_t|x_t) \quad (1)$$

10.21437/Interspeech.2023-2355



(a) *Estimating confidences using CTC softmax* (b) *Estimating confidences using CEM*

Figure 1: *Confidence estimation for CTC models. Despite having many steps in common, CEM mainly operates on logits while CTC softmax uses scalar confidence values. Oranges, green, and red regions represent the target token, target word, and word boundaries, respectively. Note that orange regions are parts of the target word.*

where $\pi = (\pi_1, \pi_2, ..., \pi_T)$ is an alignment and $\pi_t \in \mathbf{A}' = \mathbf{A} \cup \{\epsilon\}$. The set $\mathbf{A}'$ contains both alphabets and the blank token (ϵ), which is used to handle non-phonetic ambiguous cases. The function, $\mathcal{B} : \mathbf{A}' \mapsto \mathbf{A}$, maps between alignments and transcriptions by removing repeated and blank tokens.

During inference, the CTC model predicts the transcription by aggregating the most likely tokens of every timestep as shown in (2). We use CTC argmax decoding for simplicity.

$$y^* = \text{argmax}_{y^*} P(y^*|x) = \text{argmax}_{\pi:\mathcal{B}(\pi)=y^*} \prod_{t=1}^{T} P(\pi_t|x_t) \tag{2}$$

The mapping function $\mathcal{B}$, while removing the reliance on ground-truth alignments, complicates the process of obtaining token posteriors for estimating confidences. Letters with long pronunciation are continuously predicted in consecutive alignment frames and will be merged into a single character by the mapping function $\mathcal{B}$. This behavior hinders the direct use of model outputs as confidence scores because a single predicted letter can have many posteriors. For an example alignment $\pi = (c, a, a, \epsilon, t)$ predicted by the CTC model, merging consecutive posteriors for the letter a and dealing with the probability of the blank token are necessary to obtain the posterior for each individual token in the final transcription, $y = (c, a, t)$.

3. Confidences for CTC Models

This section present approaches for estimating confidences of CTC predictions. We firstly provide an algorithm for acquiring confidences using token posteriors from CTC softmax outputs. Then, we propose to compute confidence scores using CEM. Procedures for each method are displayed in Figure 1.

3.1. CTC softmax as confidences

We propose to disambiguate choices of consecutive duplicate confidences by aggregating the posteriors of every corresponding frames. Specifically, we denote logit, $z_t^k : z \in \mathbb{R}^{T\times|\mathbf{A}'|}$, as the pre-softmax hidden features for an alphabet, $k \in \mathbf{A}'$, at frame t. The alignment at frame t is the character with the highest logit, $\pi_t = \text{argmax}_k z_t^k$. For a given letter y_u in the predicted transcript $y = \mathcal{B}(\pi)$, we obtain its corresponding logit, $\hat{z}_u : \hat{z} \in \mathbb{R}^{U\times|\mathbf{A}'|}$, by applying an aggregation function on the region of the sequence z where y_u is derived from, $y_u = \mathcal{B}(\pi_{t:t+N})$. This is shown in (3) below:

$$\hat{z}_u = agg_f(z_{t:t+N}) \text{ where } \forall_{0\le n\le N} \pi_t = \pi_{t+n} \tag{3}$$

We explore min, max, and mean as an aggregation function (agg_f) for the frame-level outputs. To obtain the probability of the predicted token y_u, we apply the softmax function to the aggregated logits, $CL(y_u) = \text{S}(\hat{z}_u)^{y_u}$ where $CL(\cdot)$ is a token-level confidence function and $\text{S}(\hat{z}_u)^{y_u}$ is the softmax output at position u for the letter y_u.

We split the sequence of letter-level confidences into subsequences using the predicted word boundaries. For each subsequence, the second aggregation is applied in order to get the word-level confidences as shown in (4).

$$CW(w_i) = \text{mean}(CL(\hat{z}_{u:u+m})) \tag{4}$$

where $[u, u+m]$ is the boundary for the word w_i, excluding the boundary tokens. We acquire word boundaries using spaces predicted by the ASR model. Probabilities of blank outputs also contribute to word confidences even though they do not represent any letters in the predictions. Though these confidences are derived from argmax posteriors, our approach is also compatible with probabilities from beam search decoding.

3.2. Confidence Estimation Module

To have better calibrated confidence scores, we adopt the idea of using a learnable CEM [16, 8] for predicting confidences. CEM does neither calibrate letter nor word confidences from CTC posteriors. Instead, CEM directly computes word-level confidences using word-level features aggregated from CTC logits.

Concretely, let the word w_i comprises the letters $y_{u:u+m}$ predicted by the CTC model. We denote the logit for the word w_i as $\bar{z}_i = agg_l(\hat{z}_{u:u+m})$ where agg_l is an aggregation function. CEM takes the aggregated logits ($\bar{z}_i$), softmax of the logits ($\text{S}(\bar{z}_i)$), the predicted tokens ($\rho(w_i)$), and the number of characters within the word ($\eta(w_i)$) as shown in (5).

$$\text{CEM}(w_i) = \sigma(\theta([\bar{z}_i; \text{S}(\bar{z}_i); \rho(w_i); \eta(w_i)]) \tag{5}$$

where θ is a trainable neural network, σ is a sigmoid function, and $\rho(w_i)$ is a summation of one-hot vectors of every alphabets in the word w_i. From now on, we will use the term aggregation to refer to both agg_f and agg_l, interchangeably.

We study two different types of neural networks for CEM in order to compare the effects of contexts to word confidences. The first variant is MLP CEM (CEM_{MLP}), which independently computes a confidence for each word in the sentence. We propose to consider the whole sentence as contexts for confidence prediction by using a Transformer encoder as MLP (CEM_{Tran}).

CEM is trained using binary cross entropy loss with the aim of modeling the probability of w_i being a correct word. We follow [16] and acquire the word-level ground-truths for CEM using edit-distance between predicted and ground-truth transcriptions. Correctly aligned words are positive training samples, while any substitution and insertion errors are negative samples.

It is important to note that our method does not take deletion errors into account, similar to [16].

4. Experimental setup

This section provides detailed information about the ASR and CEM models, including their training configurations, as well as info about the datasets and evaluation metrics.

4.1. Corpora

We used English and Thai corpora for training our ASR models and CEMs. The entire 960 hours of LibriSpeech was used to train the English ASR model [27]. The English CEM was trained using the *train-clean-100* subset. As for the evaluation, we used *test-clean* and *test-other* sets as in-domain test sets and used CommonVoice [28] as the out-of-domain test set.

The Thai ASR model and CEM were trained on 150 hours of Thai Podcast [29]. Thai CEM was evaluated using a 27-hour in-domain test set and 8 hours of mock technical interview corpus. The latter corpus consisted of 6k utterances, of which 27% contains code-switching. There were 91k Thai words and 3k English words in the dataset. Pre-existing spaces were removed and word tokenization was performed using DeepCut to ensure consistent use of spaces [32].

4.2. Evaluation metrics

We adopt four common metrics for the evaluation, including normalized cross entropy (NCE), expected calibration error (ECE), area under receiver operating characteristic curve (AUROC), and area under precision-recall curve (AUPR).

NCE and ECE both measure dissimilarities between actual and predicted errors. NCE depicts the differences between entropy of confidences and word correction rate (WCR), which is the ratio between number of correctly predicted words to the total number of words. ECE estimates absolute distances between binned confidences and WCR: $\text{ECE} = \sum_i \frac{|B_i|}{N} |\text{WCR}(B_i) - \text{avgConf}(B_i)|$. We equally binned confidences into 10 bins.

AUROC and AUPR measure the capability of using confidence scores as classification thresholds for distinguishing between correctly and wrongly predicted words. Since our ASR system achieved low WER, we reported AUPR of wrongly predicted words to highlight the differences between the methods.

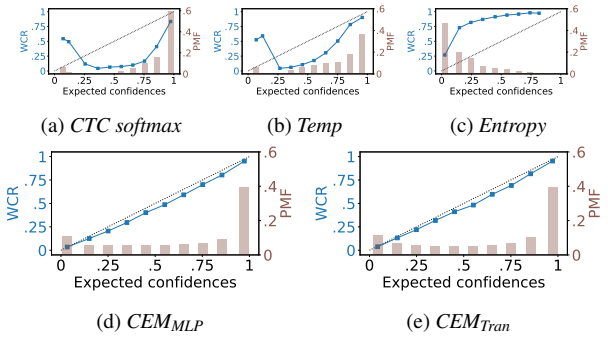
4.3. ASR models

We utilized a small Conformer-CTC model with 16 Conformer encoders and approximately 13 million parameters for our ASR system [33]. The model consumed 80-dimension filterbanks and produced character outputs. We trained the models from-scratch using AdamW optimizer [34] and Noam annealing [35] scheduler with the batch size of 128. The warm-up steps were set to 10k, and the initial learning rate was 1.0.

Our English model achieved word error rates (WERs) of 5.1%, 13.5%, and 43.3% for the test-clean, test-other, and CommonVoice, respectively. The model was trained for 200 epochs and had 29 output letters. The Thai model had 93 characters, was trained for 300 epochs, and had the WERs of 17.9% and 30.8% for Podcast and Mock Interview, respectively. We followed NeMo settings for unstated training configurations [36].

4.4. CEM

CEM_{MLP} was implemented using three feed-forward layers and Swish [37] activation function. As for CEM_{Tran}, we used a single block of a 256-dimensional Transformer encoder with a single-head attention and a dropout rate of 0.1. Both models were trained using the same data, and the best checkpoints were selected based on the AUROC and NCE of the development set. We used the mean as the frame-level aggregation for every method, following the findings in Section 5.4.



(a) *CTC softmax* (b) *Temp* (c) *Entropy*

(d) CEM_{MLP} (e) CEM_{Tran}

Figure 2: *Calibration plots for the CommonVoice test set. PMF is probability mass function.*

5. Experimental results

This section presents comparative analyses showing the effectiveness of the proposed methods and their limitations.

5.1. Confidences for English Corpora

This section presents a comparison for different confidence estimation methods, including the baseline CTC softmax (Sec. 3.1), Temperature scaling (Temp) [15], Entropy [31], CEM_{MLP}, and CEM_{Tran}. Their performances are presented in Table 1.

We found that both CEM approaches outperformed the confidences derived from CTC posteriors in every metric, indicating that they are more reliable for estimating confidence scores. CTC softmax, Temp, and Entropy performed decently as classification thresholds based on the AUROC and AUPR metrics. However, they did not accurately estimate correction rates as they struggled to achieve good NCE and ECE scores. The extreme case was Entropy which had good classification metrics, but its ECE was worse than CTC softmax.

Although CEM_{Tran} leveraged contexts, there was no significant difference in the effectiveness of CEM_{MLP} and CEM_{Tran} on in-domain test sets. However, CEM_{Tran} outperformed CEM_{MLP} on CommonVoice in all aspects. This suggests that leveraging contextual info through the attention mechanism may increase the robustness of CEM_{Tran} to handle unseen domains.

Figure 2 illustrates that the CTC model was overconfident in its softmax probabilities, even in unseen domains where it was prone to high errors. Temp showed slight improvements to CTC softmax. Entropy made underconfident scores for correct words. Both CEM methods significantly reduced discrepancies between predicted confidences and actual WCRs.

5.2. Results for Thai corpora

We studied the effectiveness of the proposed methods on Thai datasets to show the robustness across different languages and background conditions. Table 2 shows consistent improvements of CEM for Thai and English words. We found CTC softmax were excessively overconfident, and CEM_{Tran} was the best for estimating error rates of out-of-domain utterances.

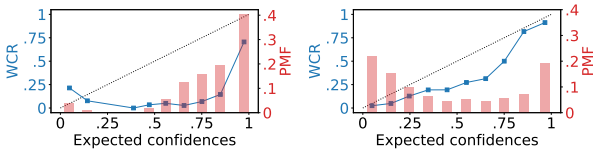
CEM_{Tran} had difficulty with rare English words in code-

Table 1: *The performance comparison for English test sets.*

	LibriSpeech test-clean/test-other				CommonVoice			
	AUROC (↑)	AUPR (↑)	NCE (↑)	ECE (↓)	AUROC (↑)	AUPR (↑)	NCE (↑)	ECE (↓)
CTC softmax	0.838/0.852	0.156/0.346	-0.433/-0.148	0.068/0.119	0.793	0.637	-0.463	0.294
temp. scaling [15]	0.859/0.866	0.181/0.365	-0.323/0.034	0.062/0.083	0.806	0.654	0.002	0.197
Entropy [31]	0.877/0.903	0.459/0.656	-4.274/-1.99	0.534/0.538	0.855	0.848	-0.450	0.394
CEM_{MLP}	0.909/0.909	0.428/0.637	0.339/0.389	0.012/0.020	0.904	0.866	0.420	0.038
CEM_{Tran}	**0.910/0.911**	**0.493/0.660**	**0.371/0.407**	**0.006/0.011**	**0.911**	**0.877**	**0.444**	**0.030**

Table 2: *The evaluation of confidence measures for Thai corpora*

	Podcast				Mock Interview			
	AUROC (↑)	AUPR (↑)	NCE (↑)	ECE (↓)	AUROC (↑)	AUPR (↑)	NCE (↑)	ECE (↓)
CTC softmax	0.805	0.326	-0.458	0.181	0.845	0.543	-0.207	0.210
temp. scaling [15]	0.815	0.339	-0.195	0.135	0.849	0.550	0.099	0.122
Entropy [31]	0.885	0.677	-1.722	0.466	**0.899**	**0.789**	-1.980	0.485
CEM_{MLP}	**0.904**	**0.701**	0.348	0.053	0.884	0.778	0.300	0.027
CEM_{Tran}	0.903	0.698	**0.360**	**0.046**	0.896	0.783	**0.390**	**0.024**



(a) *CTC softmax* (b) *CEM_{Tran}*

Figure 3: *Calibration plots for English words found in code-switching utterances presented in the Mock Interview subset.*

switching utterances of the Mock Interview test set. These English words were considered hard as they accounted for only 0.7% in the CEM training set. However, CEM_{Tran} still substantially outperformed other methods in terms of calibration errors, as displayed in Figure 3. Specifically, the ECE for English words of CTC Softmax, Temp, Entropy, CEM_{MLP}, and CEM_{Tran} were 0.507, 0.364, 0.208, 0.198, and 0.156, respectively. The AUROC values for the each method were 0.870, 0.865, 0.910, 0.892, and 0.909, respectively.

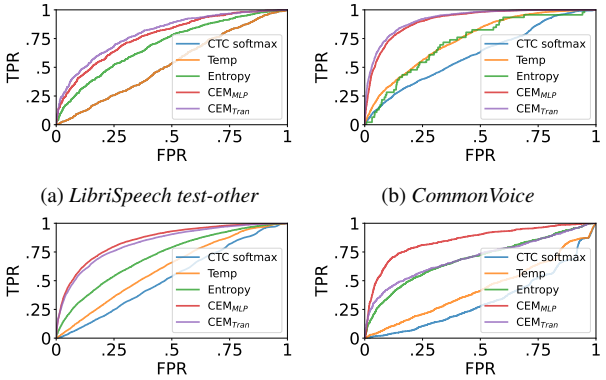
5.3. Confidences as a trigger for human intervention

One possible use case of confidence estimates is for helping human transcribers transcribe voice recordings such as video lectures or meetings. We might want to trigger a manual review if the sentence-level confidence is below a certain threshold. A simple method to estimate sentence-level confidence is to average the word-level confidence scores of the words within a sentence [8, 38]. We used correctly predicted sentences as true positives and showed the ROC curves in Figure 4. The sentence confidence estimated by the CEM models perform well in filtering out pristine transcriptions that do not require further fixes.

CEM_{Tran} exhibited the best performances for English test sets, while CEM_{MLP} showed superiority on Thai utterances. We found unexpected declines in the performances of CEM_{Tran} on Mock Interview, leaving CEM_{MLP} as the clear winner for sentence-level confidences.

5.4. Design choices

We justify our choice of frame-level aggregation (agg_f) by comparing the performances of CEM_{Tran} on LibriSpeech's test-clean. Table 3 indicates that mean aggregation tends to produce



(a) *LibriSpeech test-other* (b) *CommonVoice*

(c) *Podcast* (d) *Mock Interview*

Figure 4: *ROCs of sentence screening capability*

Table 3: *CEM_{Tran} performances on LibriSpecch test sets for different frame-level aggregation and without blank tokens*

agg_f	AUROC (↑)	AUPR (↑)	NCE (↑)	ECE (↓)
max	**0.911/0.912**	0.475/0.659	0.350/0.394	0.015/0.026
mean	0.910/0.911	**0.493/0.660**	**0.371/0.407**	0.006/**0.011**
min	0.906/0.909	0.473/0.651	0.359/0.399	**0.005**/0.013
- blanks	0.900/0.904	0.468/0.641	0.349/0.390	0.007/0.006

better results. These observations also held for CTC softmax, Temp, and CEM_{MLP}. We also reported the effect of removing blanks from the confidence estimation as they did no represent any letters in the predictions. This resulted in some degradation.

6. Conclusion

This paper introduced using CEM for predicting confidences of CTC ASR models. We presented a novel feature extraction pipeline for CEM and demonstrated significant improvements of CEM in NCE and ECE compared to using naive confidences derived from CTC softmax. We proposed using Transformer as CEM instead of MLP and shown that CEM_{Tran} was the preferred choice as it outperformed other methods in most cases. Future work includes extending CEM to handwriting recognition and investigating its use with language modeling.

7. References

[1] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in *SIGKDD*, 2015.

[2] B. Kompa, J. Snoek, and A. Beam, "Second opinion needed: communicating uncertainty in medical machine learning," *npj Digital Medicine*, vol. 4, 12 2021.

[3] D. Yu, J. Li, and L. Deng, "Calibration of confidence measures in speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, 2011.

[4] K. Veselý, M. Hannemann, and L. Burget, "Semi-supervised training of deep neural networks," in *ASRU*, 2013.

[5] K. Veselý, L. Burget, and J. Černocký, "Semi-Supervised DNN Training with Word Selection for ASR," in *Interspeech*, 2017.

[6] J. Kahn, A. Lee, and A. Hannun, "Self-training for end-to-end speech recognition," in *ICASSP*, 2020.

[7] M. N. Rizve, K. Duarte, Y. S. Rawat, and M. Shah, "In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning," in *ICLR*, 2021.

[8] D. Qiu, Q. Li, Y. He, Y. Zhang, B. Li, L. Cao, R. Prabhavalkar, D. Bhatia, W. Li, K. Hu, T. N. Sainath, and I. McGraw, "Learning Word-Level Confidence For Subword End-to-End ASR," in *ICASSP*, 2021.

[9] G. Evermann and P. Woodland, "Large vocabulary decoding and confidence estimation using word posterior probabilities," in *ICASSP*, 2000.

[10] D. Oneata, A. Caranica, A. Stan, and H. Cucu, "An evaluation of word-level confidence estimation for end-to-end automatic speech recognition," in *SLT*, 2021.

[11] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[12] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *ICASSP*, 2018.

[13] J. Li, R. Zhao, Z. Meng, Y. Liu, W. Wei, S. Parthasarathy, V. Mazalov, Z. Wang, L. He, S. Zhao, and Y. Gong, "Developing RNN-T Models Surpassing High-Performance Hybrid Models with Customization Capability," in *Interspeech*, 2020.

[14] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," 2017.

[15] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *ICML*, 2017.

[16] Q. Li, D. Qiu, Y. Zhang, B. Li, Y. He, P. C. Woodland, L. Cao, and T. Strohman, "Confidence estimation for attention-based sequence-to-sequence models for speech recognition," in *ICASSP*, 2021.

[17] M. Wang, H. Soltau, L. E. Shafey, and I. Shafran, "Word-level confidence estimation for rnn transducers," in *ASRU*, 2021.

[18] D. Qiu, Y. He, Q. Li, Y. Zhang, L. Cao, and I. McGraw, "Multi-task learning for end-to-end ASR word and utterance confidence with deletion prediction," Apr. 2021.

[19] H. Soltau, M. Wang, I. Shafran, and L. E. Shafey, "Understanding Medical Conversations: Rich Transcription, Confidence Scores & Information Extraction," in *Interspeech*, 2021.

[20] J. Salazar, K. Kirchhoff, and Z. Huang, "Self-attention networks for connectionist temporal classification in speech recognition," in *ICASSP*, 2019.

[21] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[22] J. Lee, L. Lee, and S. Watanabe, "Memory-Efficient Training of RNN-Transducer with Sampled Softmax," in *Interspeech*, 2022.

[23] W. Zhou, W. Michel, R. Schlüter, and H. Ney, "Efficient Training of Neural Transducer for Speech Recognition," in *Interspeech*, 2022.

[24] J. Michael, R. Labahn, T. Grüning, and J. Zöllner, "Evaluating sequence-to-sequence models for handwritten text recognition," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019.

[25] D. H. Diaz, S. Qin, R. Ingle, Y. Fujii, and A. Bissacco, "Rethinking text line recognition models," Apr. 2021.

[26] K. Chaudhary and R. Bali, "Easter2. 0: Improving convolutional models for handwritten text recognition," *arXiv preprint arXiv:2205.14879*, 2022.

[27] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*, 2015.

[28] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in *Language Resources and Evaluation (LRE)*, 2020.

[29] B. Naowarat, T. Kongthaworn, K. Karunratanakul, S. H. Wu, and E. Chuangsuwanich, "Reducing spelling inconsistencies in code-switching ASR using contextualized CTC loss," in *ICASSP*, 2021.

[30] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[31] A. Laptev and B. Ginsburg, "Fast entropy-based methods of word-level confidence estimation for end-to-end automatic speech recognition," in *SLT*, 2022.

[32] Rakpong Kittinaradorn, Titipat Achakulvisut, Korakot Chaovavanich, Kittinan Srithaworn, Pattarawat Chormai, Chanwit Kaewkasi, Tulakan Ruangrong, Krichkorn Oparad, "DeepCut: A Thai word tokenization library using Deep Neural Network," Sep. 2019.

[33] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Interspeech*, 2020.

[34] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, 2017.

[36] O. Kuchaiev, J. Li, H. Nguyen, O. Hrinchuk, R. Leary, B. Ginsburg, S. Kriman, S. Beliaev, V. Lavrukhin, J. Cook *et al.*, "Nemo: a toolkit for building AI applications using neural modules," *arXiv preprint arXiv:1909.09577*, 2019.

[37] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," in *ICLR*, 2018.

[38] Q. Li, Y. Zhang, B. Li, L. Cao, and P. C. Woodland, "Residual Energy-Based Models for End-to-End Speech Recognition," in *Interspeech*, 2021.