



# ASR Confidence Estimation using True Class Lexical Similarity Score

Nagarathna R<sup>1</sup>, Thishyan Raj T<sup>2</sup>, Ravi Teja Chaganti<sup>2</sup>, Vipul Arora<sup>2</sup>

<sup>1</sup>Big Data Research and Supercomputing Division; AcSIR, CSIR-4PI, India

<sup>2</sup>Department of Electrical Engineering, IIT Kanpur, India

rathna@csir4pi.res.in, thishyan20@iitk.ac.in, rraja21@iitk.ac.in, vipular@iitk.ac.in

## Abstract

Deep Neural Networks (DNN) often exhibit overconfidence, leading to poor confidence calibration in Automatic Speech Recognition (ASR) models. State-Of-The-Art (SOTA) approaches to estimate confidence are based on statistical measures or auxiliary models trained in supervised way using binary target scores, which however, fail to capture the degree of errors in substituted outputs. Continuous target score uses temporal alignment between predictions and ground truth, but are prone to inaccurate temporal alignment. To address these limitations, we propose a novel target score, True Class Lexical Similarity (TruCLeS), to train the auxiliary Confidence Estimation Model (CEM). TruCLeS is based on true class probability and lexical similarity between the prediction and ground truth. Experiments with CTC and RNN-Transducer based ASR models support its superiority against SOTA approaches. The codes are available at [https://github.com/madhavlab/2025\\_trucles\\_interspeech](https://github.com/madhavlab/2025_trucles_interspeech).

**Index Terms:** confidence estimation, automatic speech recognition, uncertainty

## 1. Introduction

Confidence estimation of predictions from Automatic Speech Recognition (ASR) model enhances their trustworthiness and supports informed decision-making in various applications such as semi-supervised learning [1], speaker adaptation [2], active learning [3], and speech translation [4]. It plays a pivotal role in sensitive domains like Alzheimer's disease and depression detection, where unreliable predictions can lead to serious consequences [5]. Applications of speech recognizers typically require word-level confidence scores [6], as word units serve as a fundamental basis for various downstream tasks such as speech translation, keyword spotting, and spoken term detection.

Neural networks generate predictions based on the highest class probability scores, which are interpreted as confidence estimates for output tokens, as proposed in [7] [8] [9]. However, this approach is constrained by the poor calibration of the ASR model [10]. To prevent the accumulation of probability mass toward the best hypothesis, techniques such as scaling the class-probability distribution and applying statistical transformations like normalization and entropy are used to mitigate overconfident bias, as discussed in [11] [6]. However, these transformations may retain the underlying skewed bias, resulting in poorly calibrated outputs.

Several studies leverage auxiliary neural networks trained with target confidence scores for confidence estimation. [12] extracts various features, including Language Model (LM) scores, attention embeddings, Connectionist Temporal Classification (CTC) scores, and class probability distributions, to train

an auxiliary Confidence Estimation Model (CEM). Similarly, [13] utilizes acoustic and LM features to train an RNN-based CEM. In [14], hypothesis embeddings from the ASR decoder layer and acoustic embeddings from the encoder layer serve as inputs to train the CEM model. Other approaches, such as those in [15], [16], and [17], extract features like attention scores, input embeddings, class probability scores, top-k probability scores, and LM features to train CEM models. [15] employs a fully connected CEM, while [16] introduces residual-energy-based models, and [17] integrates LM features with the methods proposed in [15] and [16]. Similarly, [18] trains an auxiliary model for the RNN-Transducer (RNN-T) ASR system, utilizing intermediate representations from the model as input. All these approaches utilize the Levenshtein alignment method to determine whether a predicted word is correct or incorrect. Correct words are assigned a confidence target score of '1', while incorrect words receive a score of '0'. The auxiliary CEM models are trained using these binary target scores. However, binary target scores fail to capture the degree of correctness of predicted words. In our previous work [19], we train an auxiliary model using intermediate outputs from the ASR model using a continuous target score called the Temporal Lexeme Similarity Score (TeLeS). TeLeS combines temporal similarity between words in the ground truth and hypothesis with lexical similarity between corresponding words. The word start and end timestamps, in both ground truth and hypothesis, are estimated using a forced alignment approach, which entails a computational overhead. Also, the errors in timestamp estimation would lead to errors in the target confidence values.

This work is motivated by the need for confidence estimation in ASR models, the overconfident bias in class-probability distributions, the limitations of binary target scores in representing correctness levels, and the potential deviation of continuous confidence scores due to timestamp approximation errors. We introduce a novel target score for training auxiliary CEM models, termed True Class Lexical Similarity Score, abbreviated as "TruCLeS". TruCLeS leverages true class probabilities to compute confidence scores because they quantify the measure of model uncertainty, better reflect the likelihood of correct predictions, and mitigate overconfidence by incorporating lexical similarity. We evaluate our approach on both CTC and RNN-T models. We benchmark our method against State-of-the-Art (SOTA) CEM approaches for both CTC and RNN-T architectures to demonstrate its effectiveness.

The rest of the paper is organized as – Section 2 presents the problem formulation, introduces the TruCLeS score, and outlines the learning method for the TruCLeS-based CEM applied to CTC and RNN-T models. Section 3 details the experimental setup and discusses the observed results. The final section concludes the paper and offers directions for future research.

## 2. The Proposed TruCLEs Method

### 2.1. Problem Formulation

The speech dataset  $\mathcal{D}$  comprises speech-transcript pairs  $\{(X_i, \mathbf{z}_i)\}$ , where  $X_i$  is the mel-spectrogram of a speech audio file and  $\mathbf{z}_i$  is the corresponding actual transcription. Let  $X_i = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T]$ , where each  $\mathbf{x}_t \in \mathbb{R}^D$  denotes the  $D$ -dimensional mel-spectrogram magnitude of the time frame  $t \in \{1, \dots, T\}$ . Let  $\mathbf{z} = [z_1, z_2, \dots, z_U]$  be the sequence of labels. Here,  $z_u \in \mathcal{L}$ , with  $\mathcal{L}$  being the set of tokens (which can be phonemes, characters, word-pieces or byte-pair encoding). The sequence  $\mathbf{z}$  can also be grouped into a sequence of words,  $\mathbf{w} = [w_1, w_2, \dots, w_n, \dots, w_N]$ , which can be denoted as  $w_n$ . E.g.,  $[w_1, w_2, \dots, w_N] = [[z_1, z_2], [z_3, z_4, z_5], \dots, [z_U]]$ .

An ASR model  $\mathcal{F}_\Theta$  is trained using  $\mathcal{D}' \subset \mathcal{D}$  to predict the speech transcription,

$$\hat{\mathbf{z}} = \mathcal{F}_\Theta(X) \quad (1)$$

where,  $\hat{\mathbf{z}} = [\hat{z}_1, \hat{z}_2, \dots, \hat{z}_{U'}]$  is the sequence of predicted tokens.  $\hat{\mathbf{z}}$  can be grouped to sequence of words,  $\hat{\mathbf{w}} = [\hat{w}_1, \hat{w}_2, \dots, \hat{w}_{N'}]$ , which can also be denoted as  $\hat{w}_{n'}$ .

Thereafter, an auxiliary neural network  $\mathcal{K}_\Phi$  is trained to estimate confidence  $\hat{c}_{n'} \in [0, 1]$  of word  $\hat{w}_{n'} \in \hat{\mathbf{w}}$ .  $\mathcal{K}_\Phi$  uses sigmoid non-linearity in the output and is trained in a supervised way using TruCLEs target scores. We describe the TruCLEs target score in section 2.2 and  $\mathcal{K}_\Phi$  training approach in section 2.3 and 2.4 for CTC and RNN-T ASR models respectively.

### 2.2. TruCLEs score computation

First a word-level alignment is obtained using Levenshtein Alignment function to map the predicted word sequence  $\hat{\mathbf{w}}$  to the reference word sequence  $\mathbf{w}$ . Let us denote the alignment as  $g_{n'}$ ; the index  $n'$  matches the index of  $\hat{\mathbf{w}}$  by omitting the deletions (i.e., words not predicted).  $g_{n'}$  tells if the aligned  $n$  is correct (C) or substitution (S);  $g_{n'} = 0$  for insertion.

Then, we repeat this process of Levenshtein alignment at the token level mapping  $\hat{w}_{n'}$  to the reference  $w_n$  where  $n$  is obtained from  $g_{n'}$ . This token level alignment is denoted by  $k_{n'j'}$ , where  $j'$  indexes the tokens of  $\hat{w}_{n'}$  and ignores deletions. Again,  $k_{n'j'}$  tells if the aligned  $j$  is correct (C) or substitution (S);  $k_{n'j'} = 0$  for insertion. Now, we define a token-level score  $\eta_{n'j'}$  as

$$\eta_{n'j'} = \begin{cases} p_{nj} & \text{if } k_{n'j'} \in \{C, S\} \\ 0 & \text{if } k_{n'j'} = I \end{cases} \quad (2)$$

where,  $p_{nj}$  is true class probability of token, i.e., class probability of the  $j$ th token (obtained from the ASR model), that is aligned with  $j'$  token in word  $w_{n'}$ .

Finally, we define the word-level target TruCLEs score,  $c_{n'}$  for output words  $\hat{w}_{n'}$  as

$$c_{n'} = \begin{cases} \frac{\sum_{j'} \eta_{n'j'}}{\sum_{j'} 1} \times \delta(\hat{w}_{n'}, w_n) & \text{if } g_{n'} \in \{C, S\} \\ 0 & \text{if } g_{n'} = I \end{cases} \quad (3)$$

where,  $\delta(\hat{w}_{n'}, w_n)$  is the lexical similarity between  $\hat{w}_{n'}$  and  $w_n$ , which could be either jaccard similarity, levenshtein similarity or any measure.

### 2.3. Training $\mathcal{K}_\Phi$ for CTC-ASR models

We consider a CTC-based [20] ASR model  $\mathcal{F}_\Theta$  with encoder  $e$  and decoder  $d$ . Train  $\mathcal{K}_\Phi$  using  $\mathcal{D}'$ .  $X$  is given as input to  $e$ ,

$$e(X) = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_T] \quad (4)$$

where,  $\mathbf{a}_t$  is the encoded vector. This is passed to the decoder which gives a softmax output  $[s_1, \dots, s_T]$  and hidden layer output as  $[\mathbf{h}_1, \dots, \mathbf{h}_T]$ . The final output of  $\mathcal{F}_\Theta$  is the word sequence  $[\hat{w}_{n'}]$ . Let  $t_{n'}$  be the set of indices  $t$  that map to the tokens of the word  $\hat{w}_{n'}$  excluding the blank token. Note that  $t$  indexes the output before the de-duplication and blank-removal operation.

The input to the confidence estimation model  $\mathcal{K}_\Phi$  is concatenated vectors  $([\bar{\mathbf{a}}_{n'}, \bar{\mathbf{h}}_{n'}, \bar{\mathbf{s}}_{n'}])$ , which are defined as

$$\bar{\mathbf{a}}_{n'} = \frac{\sum_{t \in t_{n'}} \mathbf{a}_t}{\sum_{t \in t_{n'}} 1} \quad (5)$$

$$\bar{\mathbf{h}}_{n'} = \frac{\sum_{t \in t_{n'}} \mathbf{h}_t}{\sum_{t \in t_{n'}} 1} \quad (6)$$

$$\bar{\mathbf{s}}_{n'} = \frac{\sum_{t \in t_{n'}} \mathbf{s}_t}{\sum_{t \in t_{n'}} 1} \quad (7)$$

Thereafter,  $\mathcal{K}_\Phi$  is trained with the pair  $([\bar{\mathbf{a}}_{n'}, \bar{\mathbf{h}}_{n'}, \bar{\mathbf{s}}_{n'}], c_{n'})$ , using shrinkage loss [19] as

$$\mathcal{L}_{shrink} = \left[ \frac{\frac{1}{N'} \sum_{n'} (\hat{c}_{n'} - c_{n'})^2 e^{\hat{c}_{n'}}}{1 + e^{\lambda(\nu - \frac{1}{N'} \sum_{n'} |\hat{c}_{n'} - c_{n'}|)}} \right] \quad (8)$$

Where,  $\lambda$  and  $\nu$  are hyper-parameters to address the data imbalance between correct and incorrect words.

The following example demonstrates the estimated confidence  $c_{n'}$  using TruCLEs, highlighting how the confidence scores of substituted words correspond to their level of correctness. The reference (ground truth) and the hypothesis (ASR output) are shown in devanagari and roman scripts, followed by word-level alignments  $g_{n'}$  and  $c_{n'}$ .

```
REF:  उन्होंने यहाँ समाजवादी जनता पार्टी राष्ट्रीय का नेतृत्व किया था
      /unhonne/ /yahaan/ /samaajavaadee/ /janata/ /paartee/
      /raashtrreey/ /ka/ /netrtv/ /kiya/ /tha/
HYP:  उन्होंने यहाँ समाजवादी जनता पार्टी राजिस्ट्रिय का नेतृत्व किया था
      /unhonne/ /yaha/ /samaajavaadee/ /janata/ /paartee/
      /raajistryi/ /ka/ /netrtv/ /kiya/ /tha/
ALIGN: ['C', 'S', 'C', 'C', 'C', 'S', 'C', 'C', 'C', 'C']
TruCLes: [0.95, 0.57, 0.90, 0.92, 0.88, 0.33, 0.96, 0.91, 0.97, 0.98]
```

### 2.4. Training $\mathcal{K}_\Phi$ for RNN-T ASR models

We consider an RNN-T based [21] ASR model  $\mathcal{F}_\Theta$  with a transcription network  $F$  and prediction network  $G$ . The outputs of  $F$  and  $G$  are added and passed through softmax nonlinearity to obtain token probabilities. The model is trained using  $\mathcal{D}'$ .  $X$  is given as input to  $F$  to obtain an embedding from the penultimate layer of  $F$  transformed using a linear layer.

$$F(X) = [\mathbf{a}_1, \dots, \mathbf{a}_T] \quad (9)$$

The final predicted tokens are denoted by  $\hat{z}_u, u = 1, \dots, U'$ .

Let  $t_u$  be the emission times, i.e.,  $t$  when the model outputs the non-blank token  $u$ . Let  $E_u = [\mathbf{a}_{t_u-k}, \dots, \mathbf{a}_{t_u+k}]^T$  be a feature matrix, where  $k$  is the temporal context window. Let  $D_u$  be the embedding from the penultimate layer of prediction network  $G$  transformed using a linear layer. We obtain the attention vectors [18] as

$$\bar{\mathbf{a}}_u = \text{softmax}(E_u D_u) \odot E_u \quad (10)$$

where  $\odot$  is elementwise multiplication with broadcasting. Now,  $[\bar{\mathbf{a}}_u, D_u]$  is used as an input to the confidence model  $\mathcal{K}_\Phi$ .

$\mathcal{K}_\Phi$  is trained to predict token-level confidence using supervised training data in the form of  $\{[\bar{\mathbf{a}}_u, D_u], \eta_u\}$ . Here,  $\eta_u$  is defined in Eq. (2). Shrinkage loss of Eq. (8) is used to train  $\mathcal{K}_\Phi$ . To obtain word-level confidence at the time of inference, we average the model estimated  $\hat{\eta}_u$  over the tokens corresponding to  $\hat{w}_{n'}$ .

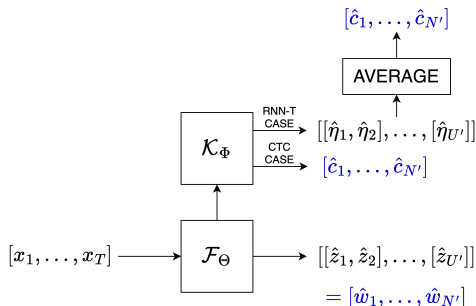


Figure 1: Block schematic of the proposed approaches

Fig. 1 shows a schematic of the proposed approaches for both CTC and RNN-T ASR models as a block diagram.

### 3. Experimental Evaluation

#### 3.1. Datasets

We assess the performance of the proposed TruCLEs based CEMs for both CTC and RNN-T based ASR models.

We utilize the pre-trained Conformer-CTC ASR NeMo model from [22], which has been trained on publicly available Hindi ASR datasets [23]. We train the TruCLEs-CEM using the KathBath (KB) [24] train dataset and evaluate it on the KB testsets. We also evaluate the TruCLEs-CEM’s performance on out-of-domain data, by using the PB Hindi dataset from [25]. The accuracy of this ASR model is given in Table 1.

For RNN-T, we utilize the pretrained Conformer-RNN-T ASR NeMo model from [26], which has been trained on publicly available English datasets. We could not find any publicly available RNN-T model for Hindi. We train the CEM using Librispeech train dataset [27]. We evaluate the performance of TruCLEs-CEM on both Librispeech test sets and (out-of-domain) Common Voice English (CVE) dataset [28]. Table 2 presents the ASR model’s performance on these datasets.

#### 3.2. CEM Implementation

For the CTC case, the CEM model architecture has three fully connected layers, with 512, 256 and 128 number of neurons across the layers, and a sigmoid layer. Rectified Linear activation function is used in the network. For the RNN-T case, we employ a network consisting of two Bi-directional LSTMs, each with 512 hidden units per direction, taking an input of dimension 2560. The Bi-LSTM layers are followed by a fully connected layer with 1024 units and sigmoid activation function. We train using learning rate 0.0001 and Adam optimizer for 100 epochs. We use the Normalized Levenshtein distance to compute lexical similarity in the TruCLEs target score. This measure normalizes the standard Levenshtein distance by the length of the strings, ensuring that the resulting value falls within the range [0,1].

Table 1: Accuracy of CTC-ASR model on KB and PB datasets

Dataset	WER	CER
KB-dev	12.75%	4.00%
KB-test sets	11.64%	3.63%
PB Hindi	18.89%	8.58%

Table 2: Accuracy of RNN-T-ASR model on Librispeech and CVE

Dataset	WER	CER
Libri-test	3.82%	1.58%
CVE-test	9.49%	3.56%

#### 3.3. Baselines

For the CTC-ASR model, we compare the results with five SOTA approaches. The approach in [7, 8, 9] uses class probabilities as the confidence of the model’s predictions. We compute the confidence of the output word by averaging the class probabilities of its tokens (excluding the blank tokens). This approach is denoted by Class Prob. The second approach, denoted by Entropy, involves non-trainable statistical measure based on exponentially normalized entropy, as proposed in [6]. The third and fourth SOTA approaches are Multi-Layer Perceptron-based CEM (denoted by CEM-MLP) and Transformer-based CEM (denoted by CEM-Trans), trained by us as described in [29]. The fifth SOTA approach is the TeLeS approach proposed in [19], which uses a continuous target confidence score for training the CEM.

For the RNN-T ASR model, we compare our method with the sub-word-based approach used in [18]. In this approach, the authors train a CEM model to estimate confidence at the sub-word level. For each word in the decoded text, the approach assigns a target score of 1 to its corresponding sub-words if the word is predicted correctly and 0 otherwise.

#### 3.4. Evaluation Metrics

We evaluate and compare our approach with SOTA using these CEM evaluation metrics - Mean Absolute Error (MAE  $\downarrow$ ), Kullback-Leibler Divergence score (KLD  $\downarrow$ ), Jensen–Shannon divergence (JSD  $\downarrow$ ), Normalized Cross Entropy (NCE  $\uparrow$ ), and Calibration Error (CE  $\downarrow$ ). The difference between the target confidence,  $C$  and the estimated confidence  $\hat{C}$  is captured by  $MAE \in [0, 1]$ . The disparity between the target score and estimated score distributions are measured by  $KLD \in [0, \infty]$  and  $JSD \in [0, 1]$ .  $NCE \in [-\infty, 1]$  [30, 31] quantifies the correlation between the correct-incorrect word distribution and the estimated confidence distribution. Two CE metrics, namely Expected Calibration Error (ECE) and Maximum Calibration Error (MCE),  $ECE \in [0, 1]$  and  $MCE \in [0, 1]$  measure the gap between the estimated confidence and accuracy of the word [32].

#### 3.5. Results

**CTC ASR case:** Tables 3 and 4 present the observed evaluation metrics for the baseline and TruCLEs CEMs on the KB-Test sets and PB hindi set. As observed in the results, Class-Prob confidence scores [7][8] exhibit poor performance due to the overconfident nature of neural networks. In con-

Table 3: Evaluation of various CTC-ASR based CEMs on KB-Test sets

Metrics	Class Prob.	Entropy	CEM MLP	CEM Trans	TeLeS	TruCLeS
MAE ↓	0.1343	0.2056	0.1811	0.2432	0.1078	<b>0.0870</b>
KLD ↓	0.4995	0.3788	0.1810	0.2938	0.1498	<b>0.1093</b>
JSD ↓	0.2792	0.1458	0.1643	0.1370	0.0430	<b>0.0278</b>
NCE ↑	-	-0.029	0.1534	0.0756	0.1408	<b>0.2971</b>
ECE ↓	0.2703	0.2418	0.1916	0.2147	0.0524	<b>0.0107</b>
MCE ↓	0.4121	0.3967	0.2072	0.2196	0.1405	<b>0.0937</b>

Table 4: Evaluation of various CTC-ASR based CEMs on PB-Hindi dataset

Metrics	Class Prob.	Entropy	CEM MLP	CEM Trans	TeLeS	TruCLeS
MAE ↓	0.1238	0.2558	0.2205	0.2042	<b>0.0917</b>	0.1005
KLD ↓	0.3925	0.2516	0.1768	0.2091	0.1887	<b>0.1627</b>
JSD ↓	0.2325	0.1258	0.2274	0.2193	0.1001	<b>0.0368</b>
NCE ↑	-	-	0.1075	0.0854	0.0913	<b>0.1919</b>
ECE ↓	0.1902	0.0199	0.2919	0.2683	0.1142	<b>0.0475</b>
MCE ↓	0.2607	0.2113	0.2919	0.2683	0.1142	<b>0.0475</b>
MCE ↓	0.3935	0.2278	0.2897	0.2183	0.2279	<b>0.1831</b>

trast, Entropy confidence measures perform better than class-probability-based scores, as they incorporate statistical transformations. However, since the underlying distribution retains the overconfidence bias, their performance is not better in comparison to other SOTA approaches. CEM-MLP and CEM-Trans train the CEM model using binary target scores that indicate whether a predicted word is correct or incorrect. The auxiliary CEM model leads to some improvement in performance compared to Class Prob and Entropy measures. However, this approach forces the CEM models to assign a confidence score of zero even for words with minor errors. Consequently, the input embeddings of such words remain close to those of correct words. This leads to poorer performance as compared to the CEMs using continuous targets. The TeLeS model [19] is trained with a continuous target score and shows better performance than the CEMs trained with binary targets. However, the temporal alignments needed to compute the target score are susceptible to inaccuracies, especially in the case of high prediction uncertainty or hallucination that result in word insertions. The superior performance of TruCLeS model supports our hypothesis that continuous target score computed directly from the true class probability and lexical similarity is better than that computed using forced temporal alignments. The performance also generalizes to out-of-domain data with a little degradation.

The confidence calibration curves in Fig. 2 further illustrate the superior performance of TruCLeS model as compared to CEM MLP model.

**RNN-T ASR case:** Table 5 presents the evaluation results for [18] and TruCLeS CEM models on the test sets. The TruCLeS model is better than the baseline across most of the metrics.

Table 5: Evaluation of RNN-T-ASR based CEM models on Libri-voice and CVE test sets

Metrics	Libri-test		CVE-test	
	[18]	TruCLeS	[18]	TruCLeS
MAE ↓	<b>0.0621</b>	0.0954	<b>0.0639</b>	0.1489
KLD ↓	0.1514	<b>0.0987</b>	0.1631	<b>0.1584</b>
JSD ↓	0.0319	<b>0.0231</b>	0.0419	<b>0.0392</b>
NCE ↑	<b>0.6901</b>	0.2604	<b>0.7011</b>	0.3012
ECE ↓	0.0961	<b>0.0118</b>	0.0259	<b>0.0191</b>
MCE ↓	0.3441	<b>0.0525</b>	0.0878	<b>0.0614</b>

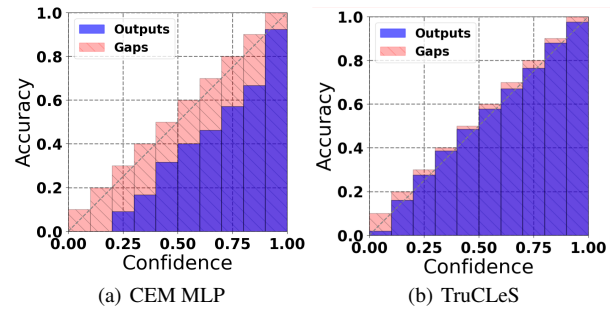


Figure 2: Calibration Curves for CTC-ASR based CEM-MLP and TruCLeS models on PB Test dataset

The calibration curves are shown in Fig. 3 showing the confidence estimated using the continuous TruCLeS target matches accuracy better than that using baseline which uses binary target.

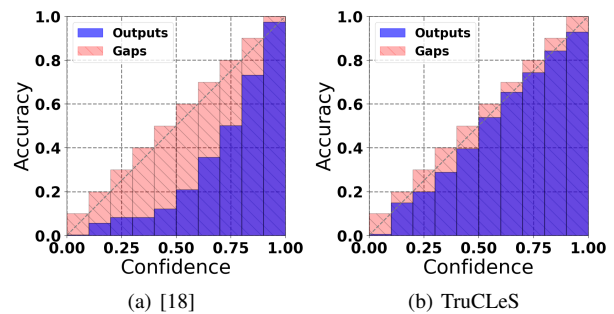


Figure 3: Calibration Curves for RNN-T ASR based [18] and TruCLeS models on Libri-test dataset

## 4. Conclusion and Future Work

In this paper, we address the limitations of existing confidence estimation methods for ASR. These limitations include the use of binary target scores and the dependence on forced temporal alignments. To overcome these limitations, we introduce the TruCLeS target score to train an auxiliary CEM. We propose ways to use TruCLeS with both CTC and RNN-T based ASR models. Experimental evaluations demonstrate the superior performance of TruCLeS-based CEM against several SOTA methods for confidence estimation in ASR.

The proposed approach helps in improving the reliability of ASR transcriptions by estimating the confidence of each predicted word. For future work, we aim to extend the system to identify deleted segments in ASR transcripts (i.e., deletions in the ASR predictions). Another promising direction is exploring semi-supervised learning for confidence modeling which will be helpful in low-resource ASR settings.

## 5. Acknowledgments

We thank Prasar Bharati for funding this project. We acknowledge the assistance of the project’s stakeholders for their consistent support and Param Siddhi for providing compute infrastructure. We also recognise the data annotation team members - Divyanshu Tripathi, Anika Kumari, Ankita Bhattacharya, Shruthi Mishra, Sachindananda Prajapathi and Shivnarayan Pandey for their meticulous effort in preparing the data.

## 6. References

- [1] D. Yu, B. Varadarajan, L. Deng, and A. Acero, “Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion,” *Computer Speech & Language*, vol. 24, no. 3, pp. 433–444, 2010.
- [2] L. Uebel and P. C. Woodland, “Speaker adaptation using lattice-based mltr,” in *ISCA Tutorial and Research Workshop (ITRW) on Adaptation Methods for Speech Recognition*, 2001.
- [3] T. Drugman, J. Pyllkkönen, and R. Kneser, “Active and semi-supervised learning in asr: Benefits on the acoustic and language models,” in *Interspeech 2016*, 2016, pp. 2318–2322.
- [4] R. Zbib, L. Zhao, D. Karakos, W. Hartmann, J. DeYoung, Z. Huang, Z. Jiang, N. Rivkin, L. Zhang, R. Schwartz *et al.*, “Neural-network lexical translation for cross-lingual ir from text and speech,” in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 645–654.
- [5] W. Wu, C. Zhang, and P. C. Woodland, “Confidence estimation for automatic detection of depression and alzheimer’s disease based on clinical interviews,” in *Interspeech 2024*, 2024, pp. 3160–3164.
- [6] A. Laptev and B. Ginsburg, “Fast entropy-based methods of word-level confidence estimation for end-to-end automatic speech recognition,” in *2022 IEEE Spoken Language Technology Workshop (SLT)*, 2023, pp. 152–159.
- [7] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, B. Li, Y. Wu, and Q. V. Le, “Improved noisy student training for automatic speech recognition,” in *Interspeech 2020*, 2020, pp. 2817–2821.
- [8] Y. Chen, W. Wang, and C. Wang, “Semi-supervised asr by end-to-end self-training,” in *Interspeech 2020*, 2020, pp. 2787–2791.
- [9] D. Hendrycks and K. Gimpel, “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” in *International Conference on Learning Representations*, 2017.
- [10] J. Zhang, V. Rajan, H. Mehmood, D. Tuckey, P. P. Parada, M. A. Jalal, K. Saravanan, G. H. Lee, J. Lee, and S. Jung, “Consistency based unsupervised self-training for asr personalisation,” in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.
- [11] D. Oneață, A. Caranica, A. Stan, and H. Cucu, “An evaluation of word-level confidence estimation for end-to-end automatic speech recognition,” in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 258–265.
- [12] A. Ogawa, N. Tawara, T. Kano, and M. Delcroix, “Blstm-based confidence estimation for end-to-end speech recognition,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6383–6387.
- [13] K. Kalgaonkar, C. Liu, Y. Gong, and K. Yao, “Estimating confidence scores on asr results using recurrent neural networks,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4999–5003.
- [14] P. Swarup, R. Maas, S. Garimella, S. H. Mallidi, and B. Hoffmeister, “Improving asr confidence scores for alexa using acoustic and hypothesis embeddings,” 2019.
- [15] Q. Li, D. Qiu, Y. Zhang, B. Li, Y. He, P. C. Woodland, L. Cao, and T. Strohmaier, “Confidence estimation for attention-based sequence-to-sequence models for speech recognition,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6388–6392.
- [16] 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*, 2021, pp. 4069–4073.
- [17] Q. Li, Y. Zhang, D. Qiu, Y. He, L. Cao, and P. C. Woodland, “Improving confidence estimation on out-of-domain data for end-to-end speech recognition,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6537–6541.
- [18] M. Wang, H. Soltan, L. E. Shafey, and I. Shafran, “Word-level confidence estimation for rnn transducers,” in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 1170–1177.
- [19] N. Ravi, T. T. Raj, and V. Arora, “Teles: Temporal lexeme similarity score to estimate confidence in end-to-end asr,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 4399–4408, 2024.
- [20] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.
- [21] A. Graves, “Sequence transduction with recurrent neural networks,” *arXiv preprint arXiv:1211.3711*, 2012.
- [22] “GitHub - AI4Bharat/indic-asr-api-backend: Indic-Conformer models for ASR — github.com,” <https://github.com/AI4Bharat/indic-asr-api-backend>, [Accessed 13-02-2025].
- [23] K. S. Bhogale, D. Mehendale, N. Parasa, S. K. R. G, T. Javed, P. Kumar, and M. M. Khapra, “Empowering low-resource language asr via large-scale pseudo labeling,” in *Interspeech 2024*, 2024, pp. 2519–2523.
- [24] AI4Bharat, “Indicsuperb: Benchmarking speech processing in indic languages,” GitHub repository, 2023, [Accessed: 2025-02-20]. [Online]. Available: <https://github.com/AI4Bharat/IndicSUPERB>
- [25] “PB Hindi ASR dataset — zenodo.org,” <https://zenodo.org/records/11162885>, [Accessed 13-02-2025].
- [26] “STT En Conformer-Transducer Medium — NVIDIA NGC — catalog.ngc.nvidia.com,” [https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/stt\\_en\\_conformer\\_transducer\\_medium](https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/stt_en_conformer_transducer_medium), [Accessed 13-02-2025].
- [27] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An asr corpus based on public domain audio books,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.
- [28] Mozilla, “Common voice: A massively-multilingual speech corpus,” Online, 2020, accessed: 2025-02-20. [Online]. Available: <https://commonvoice.mozilla.org/>
- [29] B. Naowarat, T. Kongthaworn, and E. Chuangsuwanich, “Word-level confidence estimation for ctc models,” in *Proc. Interspeech 2023*, 2023, pp. 3297–3301.
- [30] M.-h. Siu, H. Gish, and F. Richardson, “Improved estimation, evaluation and applications of confidence measures for speech recognition,” in *Eurospeech*, 1997, pp. 831–834.
- [31] “nist.gov,” <https://www.nist.gov/system/files/documents/2017/11/30/nce.pdf>, [Accessed 14-02-2025].
- [32] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in *International conference on machine learning*. PMLR, 2017, pp. 1321–1330.