Bachelor's Programme in Science and Technology

Confidence Estimation of Speech Recognition Modules Using Deep Learning

A study on confidence estimation in speech recognition using deep neural networks

Youngbin Pyo

Aalto University

BACHELOR'S THESIS

Aalto University

BACHELOR'S THESIS 2017

Confidence Estimation of Speech Recognition Modules using deep learning

Study of Deep Neural Networks in the use of Confidence Estimation for Speech Recognition

Youngbin Pyo

Thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Technology.
Otaniemi, 20 Sep 2024

Supervisor:	Luc St-Pierre
Advisor:	Silas Rech

Aalto University
School of Science
Bachelor's Programme in Science and Technology

Author
Youngbin Pyo

Title
Confidence Estimation of Speech Recognition Modules using deep learning

School School of Engineering

Degree programme Bachelor's Programme in Science and Technology

Major Computational Engineering **Code** ENG3082

Supervisor Luc St-Pierre

Advisor Silas Rech

Level Bachelor's thesis **Date** 20 Sep 2024 **Pages** 27 **Language** English

Abstract

This paper provides a general overview for confidence estimation in automatic speech recognition (ASR) systems, focusing on two state-of-the-art methods: OpenAI's Whisper and NVIDIA's NeMo framework. The goal of the study is to address challenges in ASR by improving confidence estimation modules.
The methodology involved evaluating Whisper on LibriSpeech and TIMIT datasets, measuring performance on Word Error Rate (WER). Moreoever, confidence estimation techniques were applied to the Conformer-CTC and Conformer-Trasducer models built in the NeMo framework.
The results from this paper align with previous studies done on the same methods and also demonstrate the exceptional performance of Whisper on the TIMIT dataset, with WER as low as 2.73% fo large-v1. For NeMo, proposed modification to confidence estimation methods, particularly using Gibbs entropy-based measures, showed improvements in certain metrics for RNN-T methods on the Librispeech 'test-other' dataset. This paper confirms that while Whisper and NeMo demonstrate strong performance, there is room for improvement in confidence estimation techniques.

Keywords ASR, confidence estimation, end-to-end models

urn https://aaltodoc.aalto.fi

Contents

1. Introduction

Automatic Speech Recognition has seen remarkable progress in recent years. Driven by the development of increased accuracy and compact algorithms [19] with the availability of larger datasets, key advancements were made in end-to-end models, transformer-based models, and noise-robust models. This growth has seemingly impacted various domains such as voice assistants, transcription services, and educational support. Despite significant advancements in automatic speech recognition over recent years, traditional models still present several research gaps and challenges that need to be addressed.

- Forced Alignment Problem: calculates the probability of each possible alignment from an input audio sequence and returns the most likely result [9].

- Performance dropping during speech overlap: words within speech overlap show higher perplexity [20].

- Recognition is sub-optimal for children and people with disabilities: studies have shown the lack of data on children's speech for training [21].

- Recurrent model requires a lot of training data and hallucination in language models is found frequently for large models

- The lack of precision in decoding lattices for auto-regressive decoders in end-to-end systems arises from the inability to construct compact representations [1]

One approach against these problems is improvements to the confidence estimation module for automatic speech recognition (ASR) systems which have been endorsed in many speech recognition sectors. Confidence estimation in ASR functions as a correctness evaluator, calculating the probability of an input speech text's true probability. The following paragraph illustrates an example of confidence estimation applied in a speech recognition model.

Consider a LAS sequence-to-sequence model that consists of four parts: an encoder, an attention mechanism, a decoder, and a softmax layer. A one-dimensional vector with L columns is inserted into the encoder returning a feature sequence. The information is passed onto the attention layer, applying soft alignment to the input vector. It is then decoded to predict the output symbol.

$$e_{1:L} = \text{ENCODER}(x_{1:L}) \tag{1.1}$$

$$a_t = \text{ATTENTION} a_{t-1}, d_{t-1}, e_{1:L} \tag{1.2}$$

$$d_t = \text{DECODER}(a_t, d_{t-1}, \text{EMB}(y_{t-1})) \tag{1.3}$$

The confidence scores from the output tokens at this stage of the model remain uncertain. By adding a softmax layer, the sequence of tokens representing graphemes or word pieces is output indicating probabilities of correctness likelihood for each token.

$$p(y_t|y_{1:t-1}, X_{1:L}) = \text{SOFTMAX}(d_t) \tag{1.4}$$

While utilizing a softmax layer can transform the output into values interpreted as probabilities of the input speech signal, it is challenging to guarantee that all the steps in the sequence-to-sequence model have been executed correctly. Such problems may rely on largely scaled or an autoregressive nature in the decoder state. A confidence estimation module is presented to estimate the probability of the predicted output. In the LAS model, CEM takes information from the attention, the decoder, and the softmax layer into the fully-connected layer followed by a sigmoid function to generate the confidence score.

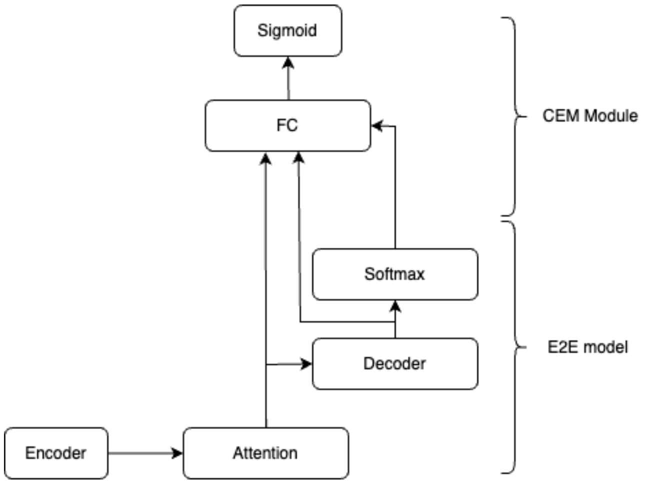


Figure 1.1. Confidence estimation module (CEM) for attention-based sequence-to-sequence models

$$p_t = \text{SIGMOID}(FC(a_t, d_t, \text{EMB}(y_t))) \tag{1.5}$$

The motivation for adding an additional confidence estimation module stems from issues inherent in a single model, such as difficulties in computing word posteriors with auto-regressive decoders, poor calibration in large-scale models, and uncertainties in execution across layers in end-to-end models. Although probabilities can be obtained from the softmax layer, it has been shown that the softmax distribution has a poor direct correspondence to confidence, as described in [23]. This thesis aims to first provide a comprehensive literature review analyzing the viability of various recognition techniques using confidence scores and to identify optimal parameters for state-of-the-art ASR models. This work done is mainly divided into two parts. First, experiments done on a model trained on a large-scope dataset is evaluated. Additionally, an ASR framework for developing generative AI is assessed to find an optimal condition for confidence estimation.

The remainder of the paper is divided into four sections. Section II reviews the different state-of-the-art methods used in speech processing. Section III describes the methodology and algorithms behind the proposed improvements in each model. Section IV provides a comprehensive review of the results from each model. Discussions and Conclusions are stated in sections V and VI, respectively. The implementation of the studies is available in the GitHub repository[1].

[1] https://github.com/ypyo01/Thesis

2. Background

The following section outlines the basics of speech processing and state-of-the-art methods in speech processing. The objective is to capture the semantics of modern speech technologies and analyze potential improvements and enhancement of such models.

2.1 Overview of Speech processing

The goal of speech processing is to facilitate communication between humans, or between humans and machines. It has numerous applications such as detecting specific words in a speech, converting an audio conversation into text, digital speech coding, and even medical analysis of speech signals [3]. For an ASR model, its purpose lies in turning speech into text. In mathematical terms, a Bayesian decision rule is employed to find the most probable word sequence, where $\hat{H}$ is the hypothesis obtained from the observation sequence O and H is the prior probability.

$$\hat{H} = argmax_H \frac{P(O|H)}{P(H)} P(O) \tag{2.1}$$

$$\hat{H} = argmax_H P(O|H)P(H) \tag{2.2}$$

$$\hat{H} = argmax_H P(H|O) \tag{2.3}$$

2.2 ASR System

The general structure of an ASR system includes input speech processing, acoustic modeling, and language modeling. While many variations exist, one popular architecture consists of feature extraction, an acoustic model, a language model, and a decoder. Feature extraction is equivalent to data pre-processing where the input data is converted into feature vectors. An acoustic model consists of statistical data on distinct sounds in a word whereas a language model contains longer vocabularies with the probability of occurrence for each word. The decoder then identifies the corresponding sounds from the models and returns the output [3].

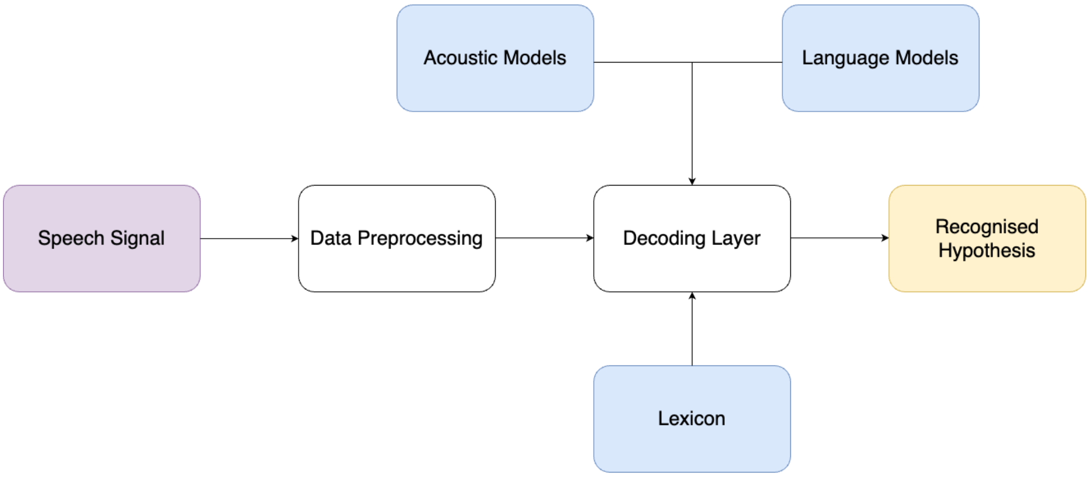


Figure 2.1. ASR System Architecture [3]

2.3 HMM-Based and E2E Model

This section presents an example of two different models that have been employed in ASR, specifically for large vocabulary speech recognition: HMM-based and End-to-End models. A HMM-based model is constructed of three parts: acoustic, pronunciation, and language model. The acoustic model calculates DNN-based posterior probabilities to identify hidden states for the acoustic model. A language model calculates the likelihood of multiple occurrences of words in a given language. Finally, a pronunciation model maps the probability calculation relationship in a dictionary to build a connection between acoustic and language sequences. The HMM-based model was the state-of-the-art method for ASR with the best recognition accuracy [3, 5]. However, the model posed challenges during the global optimization stage due to variations in its training methods, such as the Baum–Welch algorithm [18], and differences in datasets used across modules were found. To overcome this drawback, many end-to-end models were introduced. Instead of merging multiple models to a decoding step, end-to-end models replace it with a deep neural network, allowing a direct mapping of signals to label sequences. As a result, E2E models became the leaders in most ASR applications [8].

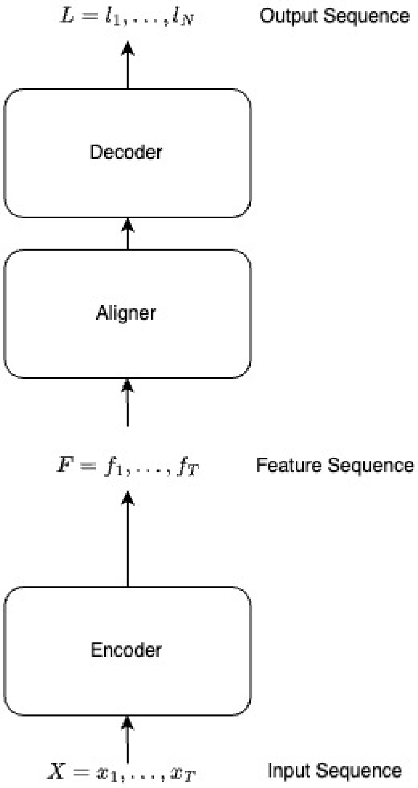


Figure 2.2. End-to-end model structure

2.4 CTC Model

Connectionist Temporal Classification (CTC) is a loss function that allows models to predict sequences of varying lengths without requiring alignment between the input and output. It assigns probabilities to all possible alignments between the input and output sequence and then sums over all alignments that can produce the target sequence [24]. It tackles limitations in the full use of DNN in normal HMM models and returns probabilities $P(y|x)$ for all possible tokens at every time step with intermediate label representations.

$$P(y|x) = \sum_{\pi \in \Phi(y')} P(\pi|x) \tag{2.4}$$

In the formula, y' is the modified output of y, leaving a 'blank' token in particular time steps to indicate gaps in the prediction.

$$P(\pi|x) \approx \prod_{t=1}^{T} P(\pi_t|x) = \prod_{t=1}^{T} q_t(\pi_t) \tag{2.5}$$

$$L_{CTC} = -\ln P(y^*|x) \tag{2.6}$$

CTC models are generally built upon Recurrent Neural Network (RNN) models. In the formula above, $q_t(\pi_t)$ represents the output layer q at time t in the softmax activation layer. CTC essentially acts as a loss function when DNN is used in speech recognition. It is the negative log-likelihood from the given reference.

$$P(y|x) = \prod_{u=1}^{|y'|} \frac{\alpha_t(u)\beta_t(u)}{q_t(y'_u)} \tag{2.7}$$

The computation of the probability distribution can be done with the forward-backward algorithm. $\alpha_t(u), \beta_t(u)$ are the forward and backward variables representing all possible prefixes and suffixes, respectively [10].

2.5 RNN-T Model

Recurrent Neural Network Transducer is similar to CTC but overcomes significant drawbacks. CTC models cannot identify correlation in the output sequence since all tokens are considered independent. Moreover, CTC does not map sequences with longer output than input. It was proposed by [11] as an improvement from CTC models by removing conditional independent features. Like in CTC outputting blank symbols, RNN-T adds blank signs to its output. The major difference, however, comes from the function of blank signs. RNN-T blank signs are used to either move to the next time frame or to release more output units from the current frame. In other words, the model either updates the original hypothesis or moves on to process the next audio sequence [12]. This gives an advantage to RNN-T for a better performance than CTC.

3. Methodology

3.1 Proposed Models: Whisper

The work done on this thesis is based on Whisper, a recently introduced series of large-scale speech models from OpenAI [9]. Currently, the model has reached an accuracy close to that of humans. It is a sequence-to-sequence Transformer model trained across various speech processing tasks. These tasks are represented as a sequence of tokens predicted by the decoder, enabling a single model to replace multiple stages of a traditional pipeline. The figure below provides a general overview of the structure of Whisper.

Whisper, trained on both English-only and multilingual datasets, uses an encoder-decoder transformer architecture that enables scalability on large, diverse datasets. When scaled up to 680,000 hours of training data, it performs exceptionally well, demonstrating significantly reduced error rates compared to previous state-of-the-art methods and improved robustness across various tasks. It also expands on weakly supervised pre-training beyond English-only speech recognition, incorporating both multilingual and multitasking capabilities. Most models employed for ASR are fine-tuned for certain datasets, dropping the accuracy once tested on different datasets. To overcome this drawback, 117,000 hours of training were done in 96 languages other than English. As discussed in [6, 7], Whisper shows significant improvements from traditional ASR models when tested on multiple languages. Testing its performance lies on a metric known as Word Error Rate (WER). It compares the generated hypothesis with the reference word sequence. Once the two sequences are aligned, the number of insertions (I), deletions (D), and substitutions (S) of tokens

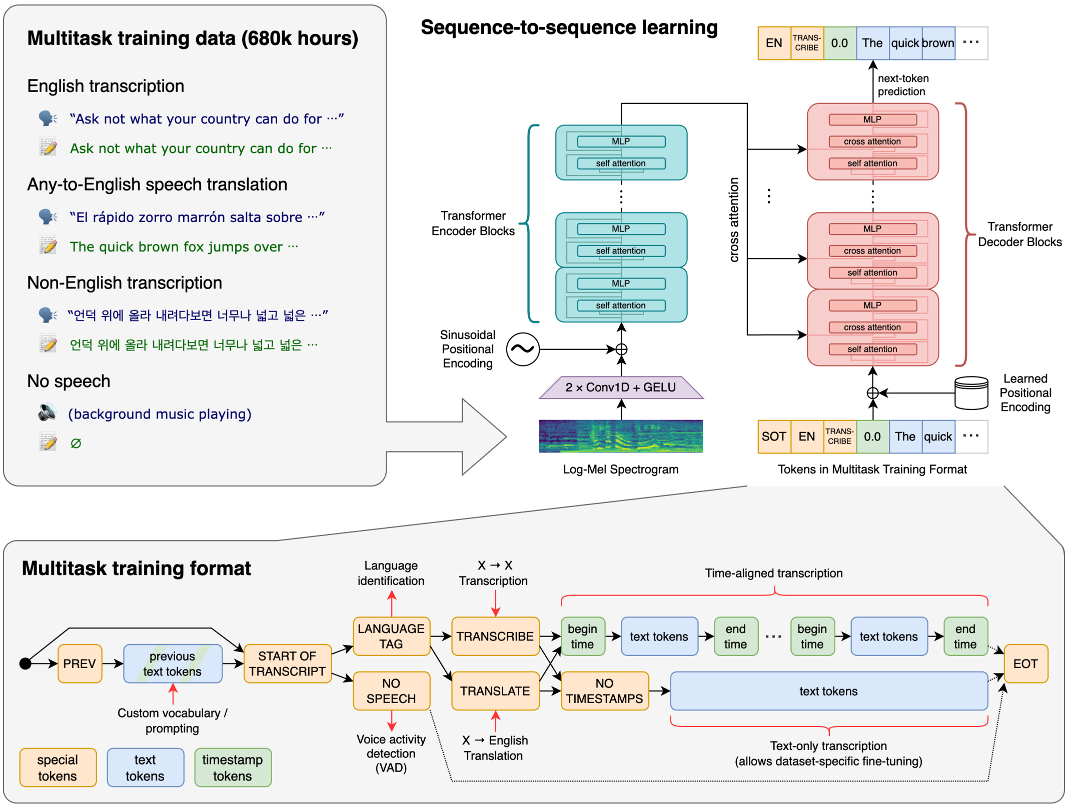

Multitask training data (680k hours)
English transcription
"Ask not what your country can do for ⋯"
Ask not what your country can do for ⋯
Any-to-English speech translation
"El rápido zorro marrón salta sobre ⋯"
The quick brown fox jumps over ⋯
Non-English transcription
"언덕 위에 올라 내려다보면 너무나 넓고 넓은 ⋯"
언덕 위에 올라 내려다보면 너무나 넓고 넓은 ⋯
No speech
(background music playing)
∅
Sequence-to-sequence learning
EN
TRANS-CRIBE
0.0
The
quick
brown
⋯
next-token prediction
MLP
cross attention
self attention
Transformer Encoder Blocks
Transformer Decoder Blocks
cross attention
Sinusoidal Positional Encoding
2 × Conv1D + GELU
Log-Mel Spectrogram
Learned Positional Encoding
SOT
EN
TRANS-CRIBE
0.0
The
quick
⋯
Tokens in Multitask Training Format
Multitask training format
Language identification
X → X Transcription
Time-aligned transcription
PREV
previous text tokens
START OF TRANSCRIPT
LANGUAGE TAG
NO SPEECH
TRANSCRIBE
TRANSLATE
begin time
text tokens
end time
⋯
begin time
text tokens
end time
NO TIMESTAMPS
text tokens
EOT
Custom vocabulary / prompting
Voice activity detection (VAD)
X → English Translation
Text-only transcription (allows dataset-specific fine-tuning)
special tokens
text tokens
timestamp tokens

Figure 3.1. Whisper architecture [4]

are summed up and are processed to get a rate to the number of errors [3].

$$WER = \frac{I + D + S}{N} * 100 \tag{3.1}$$

3.2 Proposed Models: Nvidia NeMo

This paper introduces a framework for generative AI that is used in Large Language Models (LLM), Multimodal Models (MM), Automatic Speech Recognition (ASR), Text-to-speech (TTS), and Computer Vision (CV) domains. For this paper, the performance of ASR in NeMo will be evaluated. [13] presents a confidence measure for greedy decoding, a decoding process that chooses the most likely token at each stage of generating the output sequence [25], for CTC and RNN-T-based models. An alternative to such transformer models is beam search, a system that aligns encoded feature segments simultaneously by detecting block boundaries. It then assesses how reliable each proposed interpretation is by examining end markers and recurring tokens within that interpretation [26]. Given an output estimate with the probability of the most likely signs $F(p) = \max_{v \in V} p$, here are the decoding configuration options for confidence estimation used in NeMo ASR [13]. Most metrics are based on an entropy approach due to results outperforming linear approaches in minimum and mean aggregations [5] which were the best results returned in this study. Given a probability distribution $p(v)$ across all possible tokens v, a confidence estimator that maps the predicted output can be defined as $F : P(.) \to [0, 1]$.

- $F(p) = -Hg(p) = \sum_v p_v \ln(p_v)$: Normalized probability confidence

- $F_{\text{max}}(p) = \frac{F(p)-\frac{1}{V}}{1-\frac{1}{V}} = \frac{\max p_v - \frac{1}{V}}{1-\frac{1}{V}}$: Normalized maximum probability confidence

Based on the normalized probability confidence, a base-V Shannon entropy can be applied to calculate the gibbs entropy-based confidence. Previous studies have also defined new parameters using the Tsallis and Renyi entropies. Tsallis entropy allows adjustability where smaller α leads to higher sensitivity, reducing overconfidence. Renyi entropy, like Tsallis entropy, generalizes Shannon's information theory, reducing to the base-2 Shannon entropy as its parameter α approaches 1 [13].

- $F_{\text{g}}(p) = 1 - \frac{H_{\text{g}}(p)}{max H_g(p}$: Linearly normalized Gibbs emtropy- based confidence

- $F_{\text{ts}}(p) = 1 - \frac{H_{\text{ts}}(p)}{\max H_{\text{ts}}(p)} = \frac{V^{1-\alpha} - \sum p_v^\alpha}{V^{1-\alpha} - 1}$: Linearly normalized Tsallis entropy-based confidence

- $F_r(p) = 1 - \frac{H_r(p)}{\max H_r(p)} = 1 + \frac{\log_V \sum_v p_v^\alpha}{(\alpha - 1)}$: Linearly normalized Rényi entropy-based confidence

Another approach NeMo takes is utilizing an exponential entropy and normalizing to obtain the maximum probability.

- $F^e(p) = \frac{e^{-H(p)} - e^{-\max H(p)}}{1 - e^{-\max H(p)}}$: Exponentially normalized entropy-based confidence

- $F_g^e(p) = \frac{V \cdot e^{(\sum p_v \ln(p_v))} - 1}{V - 1}$: Gibbs entropy exponential confidence measure

- $F_{\text{ts}}^e(p) = \frac{e^{\frac{1}{1-\alpha}}\left(V^{1-\alpha} - \sum p_v^\alpha\right) - 1}{e^{\frac{1}{1-\alpha}}(V^{1-\alpha} - 1) - 1}$: Tsallis entropy exponential confidence measure

- $F_r^e(p) = \frac{V(\sum p_v^\alpha)^{\frac{1}{\alpha-1}} - 1}{V-1}$: Rényi entropy exponential confidence measure

Once the model has been decoded with one of the above configurations, the confidence estimation is processed using different metrics that measures the effectiveness of a confidence estimation method. A key metric is the Area Under the Curve (AUC), which represents the probability that the model will correctly classify a randomly chosen positive sample as more likely than a randomly chosen negative sample. [22]. Commonly used Area Under the Curve of the Receiver Operating Characteristic (AUC_{ROC}) and Area Under the Precision-Recall Curve (AUC_{PR}) are implemented as well as Area Under the True Negative Rate Curve AUC_{NT} which measures the accuracy of detecting incorrect words. Moreover, Normalized Cross Entropy (NCE) is set as a confidence indicator for correct and incorrect predictions. Expected Calibration Error (ECE) is a weighted average based on the difference between absolute accuracy and confidence. Finally, three statistics on Youden's curve: AUC_{YC} (adjustability of threshold range), MAX_{YC} (optimal TNR vs. FNR tradeoff), and STD_{YC} (standard deviation of youden's curve) are given. The best value for each metric is labeled in the table below.

Metric	Best Values	Sign
AUC_{ROC}	1	Measures class separability
AUC_{PR}	1	Measures word detection rate
AUC_{NT}	0.5	Effectiveness of detecting erroneously transcribed words
AUC_{YC}	0.5	Represents the responsiveness of the model
MAX_{YC}	1	Optimal True negative rate (TNR) vs. false negative rate (FNR) tradeoff
STD_{YC}	0.25	Standard devition of YC values, indicating TNR and FNR scale differently

3.3 Dataset

Whisper was evaluated on both the LibriSpeech [15] and the TIMIT [16] dataset. The confidence estimation procedures were also tested using the Conformer-CTC and Conformer-Transducer models in the NeMo framework [14] on the LibriSpeech dataset [15] as previously done. The TIMIT database was designed through the collaboration of researchers at MIT, Texas Instruments (TI), and SRI International containing a total of 6300 samples composed of 10 sentences and 630 speakers from 8 different dialects in the United States. The LibriSpeech corpus originates from the LibriVox project, containing 16kHz audio samples up to 1000 hours.

4. Results

This thesis measures the performance of Whisper and confidence measures applied to NeMo ASR model by Nvidia. The first part of the section introduces the findings from the Whisper model. The second part presents the results from the NeMo framework.

4.1 Findings from Whisper

Table 4.1. WER values tested on TIMIT 'TEST' set with Whisper

TIMIT (TEST)	
Model	WER
base.en	5.30
small.en	3.52
medium.en	2.9
larve-v1	2.91

Table 4.2. WER values tested on LibriSpeech 'test-other' set with Whisper

LibriSpeech (test-other)	
Model	WER
base.en	10.28
small.en	7.28
medium.en	5.83
large-v1	5.55

Table 4.3. WER values tested on LibriSpeech 'test-clean' set with Whisper

LibriSpeech (test-clean)	
Model	WER
base.en	4.27
small.en	3.05
medium.en	3.02
large-v1	2.73

Tables 4.1, 4.2, and 4.3 represent Word Error Rate (WER) that resulted when Whisper was evaluated on the TIMIT database 'TEST' set and the LibriSpeech database on the 'test-other' and 'test-clean' set. Four models were used during the evaluation: base, small, medium, and large-v1.

4.2 Findings from NeMo

The following section compares the confidence metric scores obtained from the previously mentioned optimal parameter results with the newly proposed parameters. In tables 4.4 and 4.5, the first two rows are the results obtained from [13] while the third row is the best result after manual testing. Instead of targeting certain metrics, the method that performed well for all the metrics was chosen. Values labeled in bold are the leading results for each metric. α was set to 1/3 for all methods in Conformer-CTC, as well as for the mean $F^e_{ts}(p)$ and min $F^e_{ts}(p)$ in Conformer-RNN-T. For min $F^e_g(p)$, α at 1/4 had the best result. Among all the aggregation options, the minimum and mean performed the best, aligning with the results obtained from [5] when tested using entropy features.

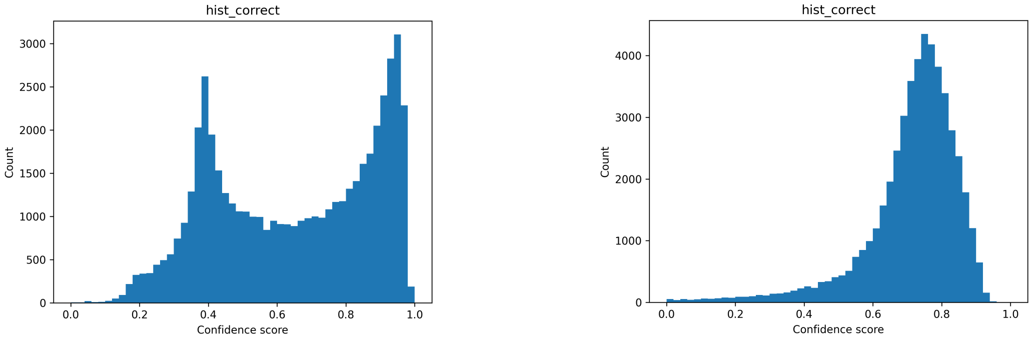
Table 4.4. Confidence metrics from CTC on LibriSpeech test-other

CTC test-other						
Method	ROC	PR	NT	YC	NCE	ECE
mean $F^e_{ts}(p)$	**90.34**	**99.21**	45.18	37.36	**0.06**	**0.08**
min $F^e_{ts}(p)$	88.04	99.04	**47.01**	**45.86**	-2.22	28.95
mean $F^e_g(p)$	88.95	99.11	40.50	30.86	-0.75	0.17

Table 4.5. Confidence metrics from RNN-T on LibriSpeech test-other

RNN-T test-other						
Method	ROC	PR	NT	YC	NCE	ECE
mean $F^e_{ts}(p)$	79.52	98.47	32.61	23.55	**-0.09**	**0.10**
min $F^e_{ts}(p)$	85.85	98.90	**47.17**	**30.71**	-2.80	0.37
min $F^e_g(p)$	**89.97**	**99.28**	40.80	36.76	-2.67	0.25

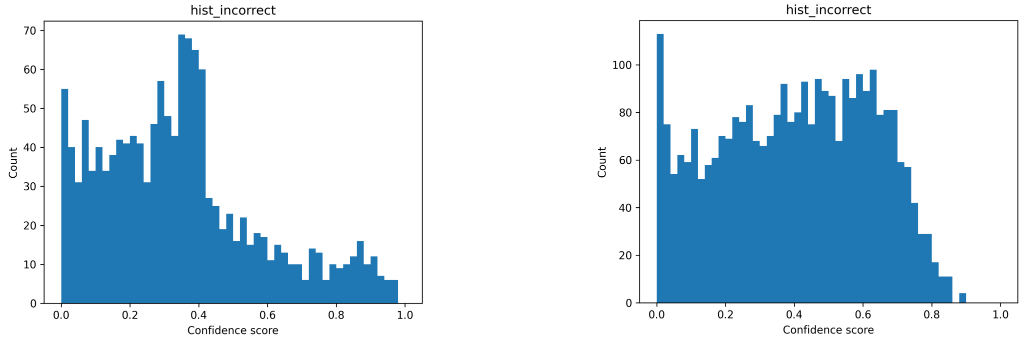
Fig. 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 compares the original results from the Conformer-CTC with the newly proposed $F^e_g(p)$, applying histograms of correctly and incorrectly recognized words as well as different confidence metrics. From Fig. 4.1 and 4.2, mean $F^e_g(p)$ clearly returns more correct and incorrect words with high confidence scores.



(a) Number of correct words using CTC on min $F_{ts}^{e}(p)$

(b) Number of correct words using CTC on mean $F_{g}^{e}(p)$

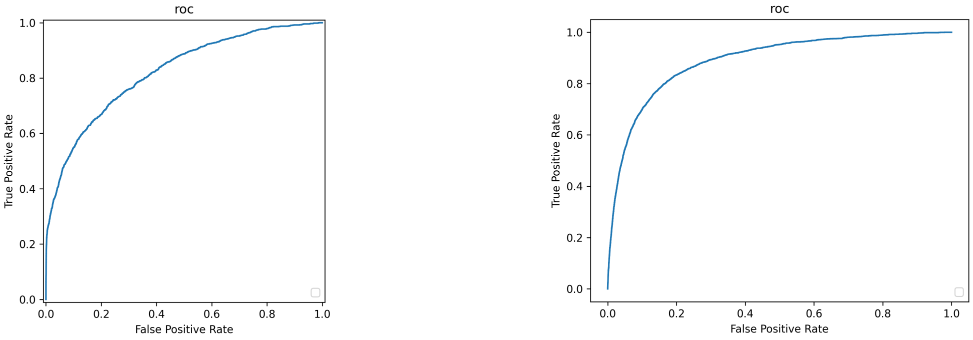
Figure 4.1. Comparison of correct words count using CTC



(a) Number of incorrect words using CTC on min $F_{ts}^{e}(p)$

(b) Number of incorrect words using CTC on mean $F_{g}^{e}(p)$

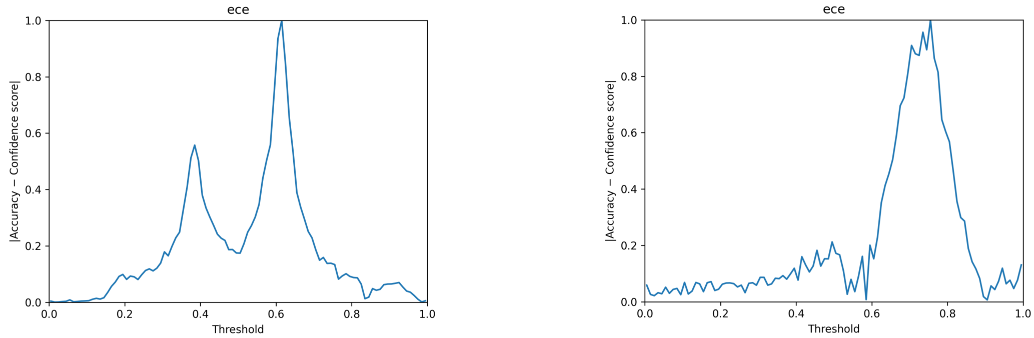
Figure 4.2. Comparison of incorrect words count using CTC



(a) AUC_{ROC} curve using CTC on min $F_{ts}^{e}(p)$

(b) AUC_{ROC} curve using CTC on mean $F_{g}^{e}(p)$

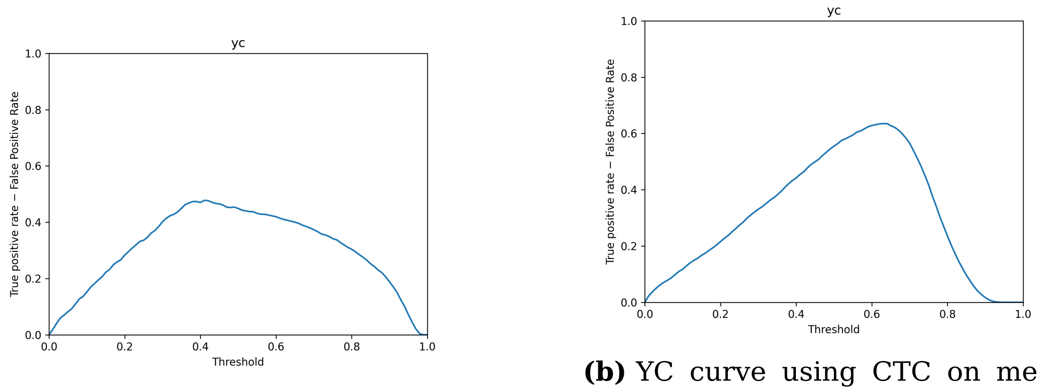
Figure 4.3. Comparison of AUC_{ROC} curve using CTC



(a) ECE curve using CTC on min $F_{ts}^{e}(p)$

(b) ECE curve using CTC on mean $F_{g}^{e}(p)$

Figure 4.4. Comparison of ECE curve using CTC on different metrics

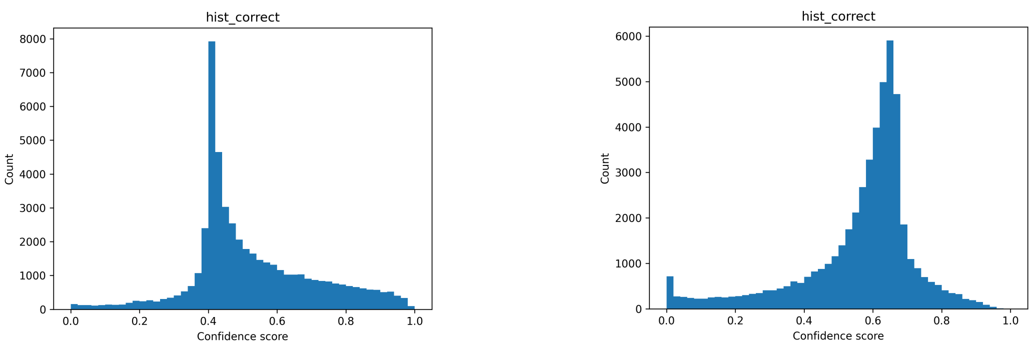


(a) YC curve using CTC on min $F_{ts}^{e}(p)$

(b) YC curve using CTC on mean $F_{g}^{e}(p)$

Figure 4.5. Comparison of YC curve using CTC on different metrics

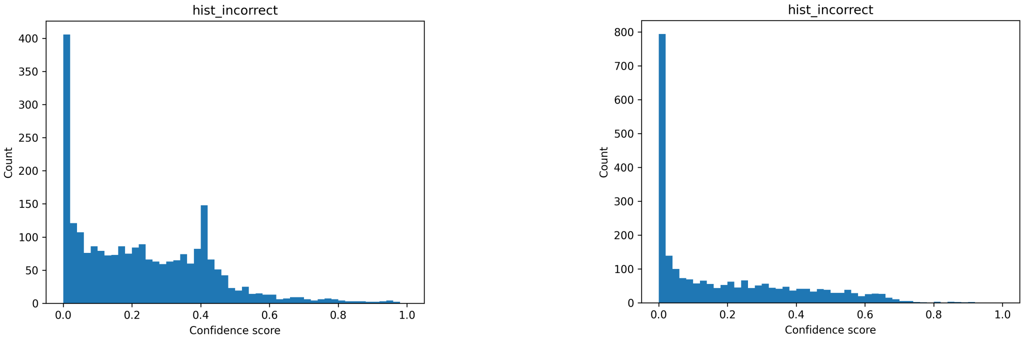
Figure 4.3 does not have significant differences between the two methods while Figure 4.4 indicates a clear distinction in threshold points where the graph spikes suggesting higher deviation between accuacy and confidence rates. min $F_{ts}^{e}(p)$ in Figure 4.5 has more data points concentrated around 0.5 contrary to mean $F_{g}^{e}(p)$ where the points have a higher variance. The performance between two methods does not widely differ.



(a) Number of correct words using RNN-T on min $F_{ts}^{e}(p)$

(b) Number of correct words using RNN-T on min $F_{g}^{e}(p)$

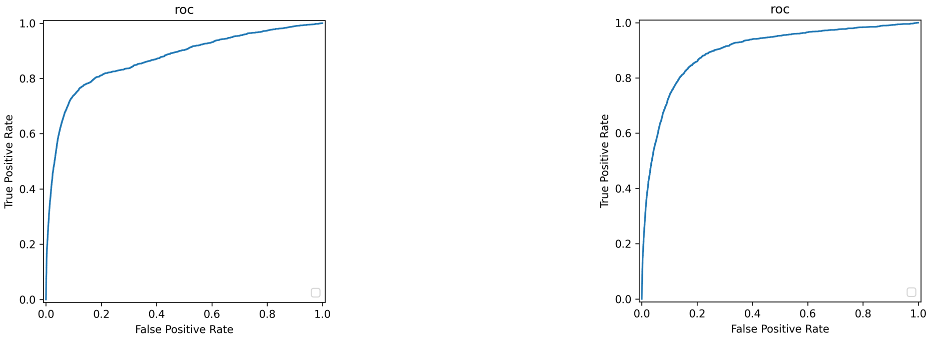
Figure 4.6. Comparison of correct word count on RNN-T



(a) Number of incorrect words using RNN-T on min $F^e_{ts}(p)$

(b) Number of incorrect words using RNN-T on min $F^e_g(p)$

Figure 4.7. Comparison of incorrect word count using RNN-T

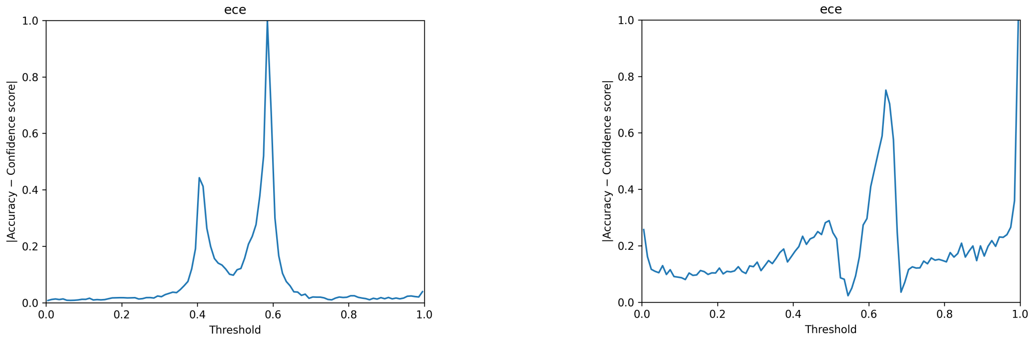


(a) AUC_{ROC} using RNN-T on min $F^e_{ts}(p)$

(b) AUC_{ROC} curve using RNN-T on min $F^e_g(p)$

Figure 4.8. AUC_{ROC} curve comparison using RNN-T

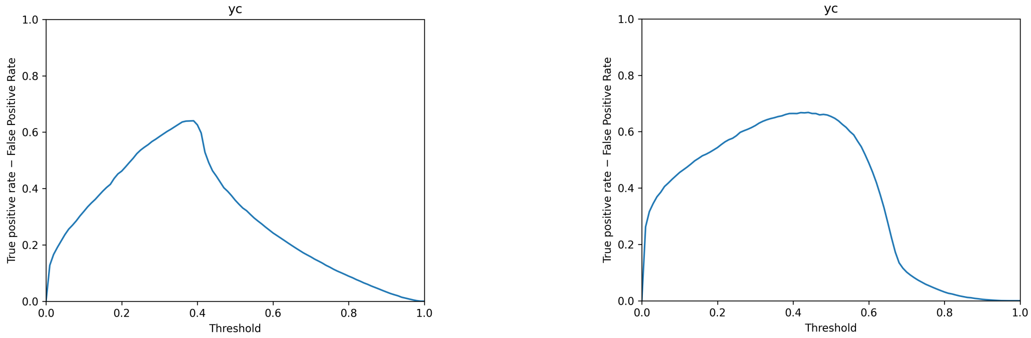
Fig. 4.6, 4.7, 4.8, 4.9, and 4.10 compares the original results from the Conformer-RNN-T with the newly proposed $F^e_g(p)$, applying histograms of correctly and incorrectly recognized words as well as different confidence metrics. The number of correct words increased in higher confidence scores and returned less incorrect words in the same confidence score points as shown in Fig. 4.6 and 4.7. The AUC_{ROC} showed a little increase in area as shown in figure 4.8. ECE values spiked around 0.6 for both methods in figure 4.9 and the $F^e_g(p)$ method spiked again close to the threshold limit.



(a) ECE curve using RNN-T on min $F^e_{ts}(p)$

(b) ECE curve using RNN-T on min $F^e_g(p)$

Figure 4.9. Comparison of ECE curve using RNN-T



(a) YC curve using RNN-T on min $F^e_{ts}(p)$

(b) YC curve using RNN-T on min $F^e_g(p)$

Figure 4.10. Comparison of YC curve using RNN-T

The AUC_{YC} area in figure 4.10 seems to be slightly larger in the $F^e_g(p)$ method despite its drop around the 0.5 threshold. The peak of the two methods are the same, but the $F^e_{ts}(p)$ has a smoother curve. Both plots demonstrate that the number of data points around the 0.5 mark is the same. Therefore, both systems performed equally on this metric.

Overall, both suggested methods for Conformer-CTC ($meanF^e_g(p)$) and Conformer-RNN-T ($minF^e_g(p)$) appeared to have improvements in guessing the number of correct and incorrect words. The plots drawn for each metric presents similar results and no clear differences were to be found. The proposed method did not perform well as the original method for Conformer-CTC, but improvements were found in ROC and PR metric in Conformer-RNN-T.

5. Discussion

Most results obtained from this research aligns well with previously obtained results. One interesting finding, though, appears in Figure 4.1 and can be explained as follows: the first peak likely results from minimum aggregation predictions clustering in the low-to-medium confidence range, while the second peak suggests an increase in correct word predictions as confidence scores rise. The metrics in NeMo appeared to have a similar trend to the orignal results where ROC, PR, NT, and YC with NCE and ECE were inversely proportional. Moreover, the increase in the number of parameters and layers for larger models in Whisper has enabled the model to perform accurately in more complicated tasks. This study has provided insights into both the excellent performance of Whisper on the TIMIT dataset and the improvements in certain metrics for RNN-T methods on the LibriSpeech test-other dataset.

Table 5.1. Whisper model architecture and details

Size	Model	Layers	Number of Parameters
base	base.en	6	72M
small	small.en	12	241M
medium	medium.en	24	762M
large	large-v1	32	1541M

6. Conclusion and Future Work

This thesis reviewed how different models and confidence estimation build ASR systems for speech-related tasks. In this work, a detailed overview of ASR and its core models have been presented. A study was conducted comparing two state-of-the-art speech recognition methods: Whisper and NeMo. The evaluation utilized two key performance metrics: Word Error Rate (WER) and NeMo's built-in confidence scores.

As demonstrated by the results, Whisper exhibits exceptional performance on tests conducted on a new database. However, the use of WER still introduces problems and will require further research on more precise metrics. WER strictly penalizes all minor differences between the hypothesis and the reference sequence [4]. Consequently, correctly recognized outputs can also be misjudged, leading to a higher error rate. To address this problem, further studies can introduce new standard metrics or apply NeMo ASR confidence metrics into Whisper. Additionally, testing the WER rate onto Whisper can be done with additional models. The models used in this study were english-only models and utilizing multilingual models may return different results. Further research with multilingual models could analyse and develop the use of numerous languages in ASR, especially low-resource languages. Whisper also has different versions of large models: large-v2, large-v3, large which can also be evaluated. Finally, testing NeMo onto the TIMIT dataset may provide new implications on its performance as it is not included in the collection of trained dataset [17].

7. References

1. Qiujia Li, David Qiu, Yu Zhang, Bo Li, Yanzhang He, Philip C. Woodland, Liangliang Cao, Trevor Strohman "Confidence estimation for attention-based sequence-to-sequence models for speech recognition: IEEE Conference Publication: IEEE Xplore," ieeexplore.ieee.org, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9414920 (accessed Aug. 15, 2024).

2. X. Shi, H. Luo, Z. Gao, S. Zhang, and Z. Yan, "Accurate and Reliable Confidence Estimation Based on Non-Autoregressive End-to-End Speech Recognition System," ISCA, https://www.isca-speech.org/archive/interspeech_2023/shi23b_interspeech.html (accessed Aug. 15, 2024).

3. Tom Bäckström, Introduction to Speech Processing: 2nd Edition. Zenodo, 2022. doi:10.5281/zenodo.6821775.

4. Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, Ilya Sutskever "*Robust speech recognition via large-scale weak supervision*", cdn.openai.org,https://cdn.openai.com/papers/whisper.pdf (accessed Aug. 16, 2024)

5. D. Oneaţă, A. Caranica, A. Stan and H. Cucu, "An Evaluation of Word-Level Confidence Estimation for End-to-End Automatic Speech Recognition," 2021 IEEE Spoken Language Technology Workshop (SLT), Shenzhen, China, 2021, pp. 258-265, doi:10.1109/SLT48900.2021.9383570.

6. L. R. S. Gris et al., "Evaluating openai's whisper ASR for punctuation prediction and topic modeling of Life Histories of the Museum of the person," arXiv.org, https://arxiv.org/abs/2305.14580 (accessed Sep. 8, 2024).

7. O. C., C. Kim, and K. Park, "Building robust Korean speech recognition model by fine-tuning large pretrained model," Phonetics and Speech Sciences, vol. 15, no. 3, pp. 75-82, 2023.

8. R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schlüter and S. Watanabe, "End-to-End Speech Recognition: A Survey," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 32, pp. 325-351, 2024, doi:10.1109/TASLP.2023.3328283.

9. E. Rastorgueva, How does forced alignment work?, https://research.nvidia.com/labs/conv-ai/blogs/2023/2023-08-forced-alignment/ (accessed Sep. 8, 2024).

10. S. Kim, T. Hori and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 2017, pp. 4835-4839, doi:10.1109/ICASSP.2017.7953075.

11. A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in Proc. ICML, Pittsburgh, PA, Jun. 2006, pp. 369–376.

12. M. Jain et al., “RNN-T for latency controlled ASR with improved beam search,” arXiv.org, https://arxiv.org/abs/1911.01629 (accessed Sep. 9, 2024).

13. A. Laptev and B. Ginsburg, "Fast Entropy-Based Methods of Word-Level Confidence Estimation for End-to-End Automatic Speech Recognition," 2022 IEEE Spoken Language Technology Workshop (SLT), Doha, Qatar, 2023, pp. 152-159, doi:10.1109/SLT54892.2023.10022960.

14. E. Harper et al., NeMo: a toolkit for Conversational AI and Large Language Models.

15. V. Panayotov, G. Chen, D. Povey and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, 2015, pp. 5206-5210, doi:10.1109/ICASSP.2015.

7178964.

16. Garofolo, John S., et al. TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1. Web Download. Philadelphia: Linguistic Data Consortium, 1993.

17. NVIDIA, “STT en conformer-CTC Medium: Nvidia NGC,” NVIDIA NGC Catalog, https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/stt_en_conformer_ctc_medium (accessed Sep. 16, 2024).

18. J. Li, J.-Y. Lee, and L. Liao, “A new algorithm to train hidden Markov models for biological sequences with partial labels,” BMC Bioinformatics, vol. 22, no. 1, Mar. 2021, doi:https://doi.org/10.1186/s12859-021-04080-0.

19. M. Picheny et al., "Trends and advances in speech recognition," in IBM Journal of Research and Development, vol. 55, no. 5, pp. 2:1-2:18, Sept.-Oct. 2011, doi:10.1147/JRD.2011.2163277.

20. O. Cetin and E. Shriberg, "Speaker Overlaps and ASR Errors in Meetings: Effects Before, During, and After the Overlap," 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, Toulouse, France, 2006, pp. I-I, doi:10.1109/ICASSP.2006.1660031.

21. Claus, F., Gamboa Rosales, H., Petrick, R., Hain, H.-U., Hoffmann, R. (2013) A survey about ASR for children. Proc. Speech and Language Technology in Education (SLaTE 2013), 26-30, doi:10.21437/SLaTE.2013-4.

22. Google. “Classification: ROC and AUC.” Google for Developers, Google, 3 Sept. 2024, developers.google.com/machine-learning/crash-course/classification/roc-and-auc#:~:text=The%20area%20under%20the%20ROC. Accessed 19 Sept. 2024.

23. D. Hendrycks & K. Gimpel, “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” *Proc. ICLR*, Toulon, 2017.

24. A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proc. ICML*, Pittsburgh, PA, Jun. 2006, pp.

369–376.

25. IBM, “Foundation model parameters: decoding and stopping criteria,” www.ibm.com, Jun. 19, 2024. https://www.ibm.com/docs/en/watsonx/saas?topic=lab-model-parameters-prompting (accessed Sep. 19, 2024).

26. E. Tsunoo, Y. Kashiwagi and S. Watanabe, "Streaming Transformer Asr With Blockwise Synchronous Beam Search," 2021 IEEE Spoken Language Technology Workshop (SLT), Shenzhen, China, 2021, pp. 22-29, doi: 10.1109/SLT48900.2021.9383517.