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

# Confidence measures in encoder-decoder models for speech recognition

*Alejandro Woodward, Clara Bonnín, Issey Masuda, David Varas, Elisenda Bou-Balust, and Juan Carlos Riveiro*

Vilynx

alejandro@vilynx.com, clara@vilynx.com, issey@vilynx.com, david.varas@vilynx.com, eli@vilynx.com, jc@vilynx.com

## Abstract

Recent improvements in Automatic Speech Recognition (ASR) systems have enabled the growth of myriad applications such as voice assistants, intent detection, keyword extraction and sentiment analysis. These applications, which are now widely used in the industry, are very sensitive to the errors generated by ASR systems. This could be overcome by having a reliable confidence measurement associated to the predicted output. This work presents a novel method which uses internal neural features of a frozen ASR model to train an independent neural network to predict a softmax temperature value. This value is computed in each decoder time step and multiplied by the logits in order to redistribute the output probabilities. The resulting softmax values corresponding to predicted tokens constitute a more reliable confidence measure. Moreover, this work also studies the effect of teacher forcing on the training of the proposed temperature prediction module. The output confidence estimation shows an improvement of -25.78% in EER and +7.59% in AUC-ROC with respect to the unaltered softmax values of the predicted tokens, evaluated on a proprietary dataset consisting on News and Entertainment videos.

**Index Terms**: speech recognition, encoder-decoder, confidence

## 1. Introduction

Automatic Speech Recognition (ASR) systems are widely used for downstream tasks such as voice assistants, metadata extraction and sentiment analysis among others. The speech signal usually contains a great amount of information that may be used for these tasks. Thus, the capacity of the ASR module to generate accurate transcriptions has a critical impact in their performance. Moreover, it is paramount to know whether the transcript can be trusted or not. Generating metadata from wrong transcripts can lead to poor recommendations, wrong intent classification, or unwanted results in a query search. Confidence scores provide a measure for the reliability of the hypothesis given by ASR systems and can be used to accurately discriminate those parts of the transcript that are less likely to contain errors. This ultimately improves the predictions of the downstream tasks.

For classical ASR systems, confidence measure computation methods have been typically divided in three categories [1][2]. In the first category, a model (binary classifier) is trained to predict a confidence score from features generated by the ASR system. The second category includes posterior based approaches, in which probabilities of decoding lattices or n-best lists are used to directly compute confidence scores [3]. Finally, the utterance verification approach consists of a discriminative training procedure that attempts to adjust the parameters of the null hypothesis and an alternate hypothesis model [4]. This technique is based on a hypothesis testing procedure known as the likelihood ratio test (LRT). While confidence measures for classical ASR systems have been thoroughly studied, different approaches are still needed towards achieving reliable confidence measures in attention based encoder-decoder ASR outputs.

For modern classification using Deep Neural Networks (DNN) - including encoder-decoder architectures - the output probabilities of the model are often directly interpreted as confidence measures. Nonetheless, DNN architectures have been proven to be overconfident [5]. Due to this, these output probabilities cannot be reliably considered as an indicator of how likely is the network of being mistaken. In order to improve the reliability of output probabilities, different techniques have been proposed. Label smoothing [6] has recently shown to reduce the overconfidence of output probabilities from neural network models [7]. Other methods like temperature scaling [5] also help to ameliorate this problem. Furthermore, techniques like [8] use neural features from a frozen model to train a separate module that outputs confidence scores per sample.

Confidence measurement in attention based encoder-decoder networks has been previously studied for the purposes of translation [9] and semantic parsing [10], but is still unexplored for ASR. In this work, we present a method to obtain a reliable confidence measure in encoder-decoder based systems for ASR.

The main contribution of this work is a novel method to obtain the confidence score for each predicted token. The proposed approach consists in using internal neural features of a frozen ASR model to train a simple yet effective independent module to predict a softmax temperature value. This value is computed in each decoder time step and multiplied by the logits in order to redistribute the output probabilities without changing the accuracy of the model. Since teacher forcing is widely used in ASR systems, a secondary contribution involves the study of its effect on the training of the confidence module - which has not been considered until this work. Furthermore, it is shown that the unbalanced distribution of errors - there are more correct tokens than incorrect - also affects the temperature predictor module.

The paper is organized as follows: Section 2 describes related work on confidence estimation, Section 3 describes our attention baseline model along with its training scheme. This section also motivates the need for improvement based on our baseline and previous work. Section 4 explains our proposed method and the key differences with other approaches. Section 5 explains the different experiments made to assess the effectiveness of our method. Finally, Section 6 gives the final remarks on the paper.



http://dx.doi.org/10.21437/Interspeech.2020-2215

## 2. Related Work

Many of the recent work on confidence scores for speech recognition can be classified as model-based confidence estimators. In [11], instead of the typically used feed forward networks, recurrent neural networks (RNNs) are introduced to compute confidence scores from lattice derived features. Following the same idea, [12] and [13] use Bidirectional RNNs in order to use past and future context to make confidence predictions. Furthermore, in [13], word embeddings are used as an additional feature. More recently, [14] and [15] introduce the use of Lattice RNNs for the same task. These works do not use neural features to compute confidence scores as they are aimed for classical ASR systems.

Probability outputs of DNNs are usually interpreted as a measure of confidence in its prediction. However, these estimates tend to be overconfident as studied in [16] [5]. A large number of methods have been proposed to ameliorate this issue. One of the most popular methods to reduce overconfidence is a regularization technique named Label Smoothing [6]. This method simply substitutes the "hard" target for a mixture of the target and a uniform distribution. Furthermore, Temperature Scaling has been shown to alleviate the overconfidence problem in [5] for image classification. In their work, softmax temperature is treated as a hyperparameter and optimized using the Negative Log Likelihood objective used for training the model.

On the other hand, Bayesian Deep Neural networks try to solve this problem using a totally different approach. Bayesian Deep Learning basically tries to estimate the uncertainty of the model by using probabilistic inference approximation techniques such as Markov Chain Monte Carlo (MCMC) [17] or variational Bayesian methods [18]. Nevertheless, these type of methods have strong drawbacks in terms of training time [19] or inference time [20].

For the specific case of DNNs for end-to-end ASR models, shortly after the introduction of attention based models [21][22], other works studied some of their shortcomings. In [23] it is shown that both Label Smoothing and constant softmax temperature reduce overconfidence but, in this case, specifically for attention based encoder-decoder networks. They also show how this improves beam search decoding performance.

More recent work on confidence measures for attention based encoder-decoder networks have explored other fields such as translation [9] and semantic parsing [10]. In [9], the poor performance of the confidence measure is attributed to the End Of Sequence (EOS) output distribution and attention uncertainty. Based on this, they propose a parametric model that predicts a softmax temperature value for every token as a function of input coverage, attention uncertainty, and token probability.

## 3. Baseline confidence model

The baseline ASR system is an attention based encoder-decoder neural network [21] [22]. The encoder takes an audio input sequence $X = [x_1, \ldots, x_{T'}]$ which encodes into a hidden representation $H = [h_1, \ldots, h_T]$. The decoder then generates a sequence of tokens $Y = [y_1, \ldots, y_U]$ from the encoded representation $H$. Each output token $y_u$ is computed as a function of the state of the decoder $q_u$, the previous token $y_{u-1}$ and the context vector $v_u$ generated by the attention mechanism.

$$P(y_u|X, y_{<u}, \theta) = softmax(f(v_u, q_u)) \tag{1}$$

The decoder state $q_u$ summarizes the output sequence until time step $u$ using a recurrent neural network. On the other hand, the context vector $v_u$ is the result of a weighted sum over the encoded sequence representations. The weights are computed for each output time step $u$ using the decoder state $q_u$ and the encoded sequence $H$ by means of the scaled dot product attention [24] mechanism. The final output sequence probability distribution can be defined as:

$$P(Y|X) = \prod_{u=1}^{U} P(y_u|X, y_{<u}, \theta) \tag{2}$$

During training, negative log likelihood (NLL) is minimized to find the model parameters $\theta$ for all the elements in the batch $B = \{(x_i, y_i)\}$:

$$NLL(\theta) = -\sum_{i \epsilon B} \sum_{u=1}^{|y_i|} logP(y_u|X, y_{i,<u}, \theta) \tag{3}$$

In order to speed up computation in the decoder, we remove sequential computations following [25]. Consequently, random sampling is used instead of scheduled sampling to reduce exposure bias [26]. Label smoothing [7] is used as regularization technique, leading to increased accuracy and less overconfident output probabilities. Nevertheless, we show how the quality of the softmax output - the probability distribution - can still be greatly improved. We refer to the value of the softmax corresponding to any given token as their confidence measure.

Our baseline also includes a mechanism to prevent the premature prediction of the EOS token. This is an important characteristic of the baseline that affects the experiment involving our implementation of [9]. As mentioned in the previous section, [9] show that the EOS output probability of the encoder-decoder model affects the overall performance of their confidence measure. They reduce the EOS token probability when there is part of the input sequence that has not been yet attended. In the case of speech recognition there is no need to attend all the input sequence - some of it may correspond to silences. Instead, we propose to lower the EOS token probability when there is still speech to be attended. The presence of speech signal is determined by a Voice Activity Detector (VAD). Finally, we check - in every decoder time step - the amount of detected audio there is from the current position of maximum attention to the end of the audio.

## 4. Methodology

In this section we first introduce our proposed method to improve the confidence measure of an attention based encoder-decoder network for ASR. In the second part, we explain how teacher forcing can affect the training process of our temperature predictor and we propose a method to mitigate its effects.

### 4.1. Predictive softmax temperature

A constant temperature has limited benefits in the quality of the confidence measure and leads to the same smoothing/sharpening effect on the output probability distribution of all tokens. This may be insufficient as some output probability distributions may need to be smoothed while others should be sharpened. Instead of using a fixed temperature, we propose to use internal neural features from the ASR model to predict this value at each decoder time step. This gives our method more flexibility to increase or decrease the entropy of the output distribution of the ASR model depending on information obtained from each specific input.

The features used to predict the temperature $T$ result from the concatenation of the context vector $v_u$ and decoder state $q_u$:

$$d_u = [v_u, q_u] \tag{4}$$

On the one hand, the context vector $v_u$ contains the audio features of the attended location at time step $u$. On the other, the decoder state $q_u$ contains a summarized representation of the output tokens generated until time step $u$. As a consequence, $d_u$ contains phonetic information from the relevant audio location and grammatical information from the transcript generated until that point.

We use a deep feed-forward neural network to predict the temperature value from $d_u$. The ASR model is frozen while this separate module is trained. For computational stability reasons, the temperature predictor computes the inverse of the temperature for each token $T_u^{-1}$, resulting in the following equation:

$$T_u^{-1}(d_u) = max(0, DNN(d_u)) \tag{5}$$

The weights of the module are optimized using the same Negative Log Likelihood criteria used to optimize the weights of the encoder-decoder neural network:

$$NLL(\theta) = -\sum_{i \epsilon B} \sum_{u=1}^{|y_i|} log(softmax(l_u * T_u^{-1}(d_u))\,[y_{iu}]) \tag{6}$$

where $B$ is the set of samples in the minibatch, $l_u$ are the logits of the output layer of the ASR model and $T_u^{-1}(d_u)$ is the temperature predicted in that time step. We refer to the temperature predictor as our confidence module.

### 4.2. Effect of teacher forcing

During training, when teacher forcing is applied, most of the time the token from the ground truth is fed to the decoder instead of the token generated in the previous step. When there is a mismatch between ground truth and the generated sequence, the decoder enters a state that is never seen in inference. We observe that this mismatch causes the entropy of the output probability to increase dramatically. Since the temperature predictor uses the decoder features, we hypothesise that if teacher forcing is also used to train this module, it may harm its performance.

In order to investigate this phenomenon, we propose to decode the sequences *before* training the temperature predictor. As a consequence, the decoded sequence should be aligned to the reference transcript. In order to be able to use the Negative Log Likelihood criterion (Equation 3) to train the temperature predictor in this case, we need a reference token for every decoded token. The alignment process generates three types of errors: substitutions, deletions and insertions. While substitutions and deletions are straightforward, in the case of insertions the ground truth has no corresponding token. Since we need a generated token for every decoded one, in this case we associate the inserted token to their closest correct token in the reference. An example of this procedure is shown in Figure 1.

## 5. Experiments and results

In this section we first explain the details about the training procedure of our ASR model and confidence module. Then, we present the evaluation metrics used to compare different confidence estimation approaches. In the last part of this section, we discuss the results of the experiments.

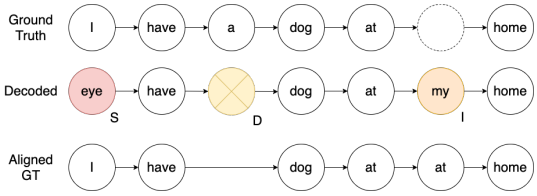


Figure 1: *Example of Aligned Ground Truth after comparing the Reference with the Decoded sequence. The three possible errors are: Substitution (S), Deletion (D) and Insertion (I).*

### 5.1. ASR model

We use an encoder-decoder model with scaled dot-product attention as our ASR model. The encoder consists on three convolutional layers as described in [27], followed by 3 bidirectional-LSTM layers [28] with 1024 units and residual connections between each layer. One last fully connected layer in the encoder outputs the final representation $H$ of size 2048. The decoder and attention methods are designed following [25] in order to remove sequential computations. The number of units in the decoder LSTM is also 1024. We choose Adam [29] as optimizer and an initial learning rate of $10^{-3}$ which is halved when the loss plateaus. The dataset used to train this model is composed of around 6000 hours of audio from news and entertainment proprietary videos.

### 5.2. Confidence module

The confidence module is composed of two fully connected layers, 1024 units per hidden layer, and a final fully connected layer with a single output. ReLU activations are used for all layers. This module is trained using Adam optimizer [29] in the same fashion as the ASR model. The training data is composed of 2000 hours of audio, which correspond to a subset of the complete ASR training dataset. The held-out test set includes 15 hours of audio.

In order to evaluate the performance of the confidence measure, the ground truth is generated in the following way. First, the decoded sequence is computed by the ASR model and aligned to the reference transcript. Then the label "1" is assigned to correct tokens whereas incorrect ones are labelled with "0". We refer to these labels as $c = c_1, \ldots, c_N$ where N is the total number of tokens in the test dataset. Note that the objective for the confidence score produced by the model is to be as similar as possible to these labels.

### 5.3. Evaluation metrics

We assess our work with the following commonly used evaluation metrics: the Normalized Cross Entropy (NCE), the Area Under Curve (AUC) of the Receiver Operating Characteristic (ROC) curve and the Equal Error Rate (EER).

Normalized Cross Entropy measures the relative change in the uncertainty of a token caused by replacing the empirical estimate of ASR correctness ($p_0$) with the predicted confidence [2][15]. If the model perfectly predicts the correct words, then the maximum value is obtained NCE = 1, however if model provides no additional information, then NCE = 0.

$$NCE = \frac{H(p_0 \cdot o, c) - H(c, \hat{c})}{H(p_0 \cdot o, c)} \tag{7}$$

$$H(p_0 \cdot o, c) = -(p_0 log(p_0) + (1 - p_0) log(1 - p_0)) \tag{8}$$

$$H(c,\hat{c}) = -\frac{1}{N}\sum_{i=1}^{N}(c_i log(\hat{c}_i) + (1-c_i)log(1-\hat{c}_i)) \quad (9)$$

where $\hat{c}$ is a vector with the confidence prediction for each token $i$ and $o$ is a vector of ones of length $N$. We use the softmax value of the predicted token as confidence score $\hat{c}_i$.

Note that it is not necessary for the prediction scores to match the correct label exactly. Instead, it is sufficient that the confidences of correctly and incorrectly decoded words are easily separated using a given threshold. For this reason, a more appropriate metric would be the ROC curve. The ROC curve shows the False Positive Rate (FPR) and True Positive Rate (TPR) computed for different confidence threshold values. Two interesting measures for the assessment of the estimated confidence derived from this curve are the AUC and the EER. The AUC values ranges from 0 when all classifications are incorrect to 1 for a perfect classifier. The EER is the point in the ROC curve where the condition $FPR = FNR = 1 - TPR$ is met. The optimal $EER = 0$ is reached when the number of False Positives and False Negatives is zero.

### 5.4. Results

In this subsection, we discuss the results of the proposed methods. The *Baseline* confidence measures are the unaltered softmax values of the predicted token. For the rest of the experiments, the confidence score is the resulting softmax value after applying the temperature scaling factor. The difference between them is the approach used to compute the temperature value for each token.

First, we evaluate how a constant temperature value affects the performance of the confidence measure (*Constant T*). Its value is set using the same dataset for training the confidence modules. As it can be seen in Table 1, there is a limited improvement in the AUC-ROC curve with respect to the *Baseline*.

Furthermore, we train and evaluate a module using the approach proposed in [9]. In this case, the performance degrades for all the metrics, and it especially harms the NCE. It is important to take into account that this method is trained while using teacher forcing [9] but evaluated over decoded sequences. In contrast, [9] evaluate their method still using teacher forcing. This could account for the lower performance seen in Table 1. Also, our *Baseline* already solves the EOS token issue (as explained in Section 3). We can conclude that the attention entropy and logits alone are not sufficient to predict a robust measure of confidence.

Our approach based on temperature prediction from decoder features trained over decoded sequences (*Decoder features*) greatly improves the three metrics: +20.35% in NCE, -21.94% in EER and +6.58% in AUC-ROC. As discussed in Section 4.2, we observe that the NCE slightly decreases with respect to the *Baseline* when teacher forcing is used (*Decoder features (TF)*). This confirms our hypothesis that teacher forcing affects the performance of the confidence module.

During experimentation, we observed a class imbalance of 10 correct confidence labels for each incorrect one. We designed an additional experiment in which a sub-sampling technique is applied in order to reduce the effect of such imbalance. We refer to this experiment as *Balanced* in Table 1. As it can be observed, the AUC-ROC and EER metrics improve, but NCE is slightly degraded - still outperforming the *Baseline*. When this technique is used with teacher forcing (*Balanced (TF)*) there is no major variation with respect to *Decoder features (TF)*.

Finally, Figure 2 shows the ROC curve for the proposed methods. Compared to the *Baseline*, the decoded feature based methods have substantially higher True Positive Rate in the range of low False Positive Rate. We observe that teacher forcing plays a major role in the training of the confidence module.

| Method | EER | AUC-ROC | NCE |
|---|---|---|---|
| *Baseline* | 0.1846 | 0.8687 | 0.4224 |
| *Constant T* | 0.1807 | 0.8720 | 0.3922 |
| *Kumar et. al* [9] | 0.1920 | 0.8668 | 0.2529 |
| *Decoder features* | 0.1441 | 0.9259 | **0.5084** |
| *Decoder features (TF)* | 0.1697 | 0.8933 | 0.4128 |
| *Balanced* | **0.1370** | **0.9347** | 0.4652 |
| *Balanced (TF)* | 0.1734 | 0.8899 | 0.4195 |

Table 1: *Results of confidence estimation methods in terms of the Equal Error Rate (EER), Area Under the Curve of the Receiver Operating Characteristic curve (AUC-ROC) and Normalized Cross Entropy (NCE). The last four rows of the table are proposed in this work.*

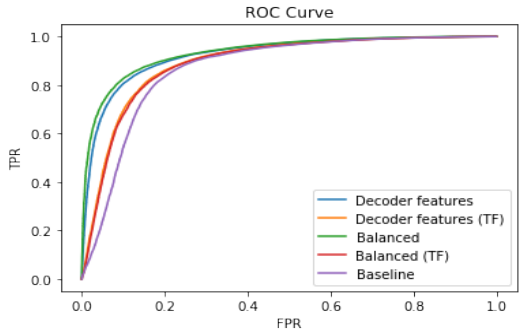


Figure 2: *ROC Curve for the proposed methods*

## 6. Conclusions

In this work, we proposed a novel method to compute a reliable confidence measure for each predicted token in attention-based encoder-decoder networks for ASR. Our approach, which is based on softmax temperature scaling, predicts a temperature value per token from the decoder internal neural features. The inverse of the temperature is then multiplied by the logits of the ASR model to readjust the values of the output softmax distribution. As a result, the new softmax values associated to each output token can be interpreted more reliably as a confidence measure. In addition, after studying the effect of teacher forcing upon the temperature predictor training, we find that it has a negative impact and therefore it is important to train the system over decoded sequences. We assess our method and report improvements of +20.35% in NCE, -21.94% in EER and +6.58% in AUC-ROC with respect to the baseline, which consists on using the unaltered softmax values of the predicted tokens. A sub-sampling technique further improves the EER and AUC-ROC metrics, achieving -25.78% in EER and +7.59% in AUC-ROC compared to the baseline.

# 7. References


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

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

[3] F. Wessel, K. Macherey, and R. Schluter, "Using word probabilities as confidence measures," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, vol. 1. IEEE, 1998, pp. 225–228.

[4] R. C. Rose, B.-H. Juang, and C.-H. Lee, "A training procedure for verifying string hypotheses in continuous speech recognition," in *1995 International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 1995, pp. 281–284.

[5] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1321–1330.

[6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[7] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" in *Advances in Neural Information Processing Systems*, 2019, pp. 4696–4705.

[8] Y. Shi and A. K. Jain, "Probabilistic face embeddings," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6902–6911.

[9] A. Kumar and S. Sarawagi, "Calibration of encoder decoder models for neural machine translation," *arXiv preprint arXiv:1903.00802*, 2019.

[10] L. Dong, C. Quirk, and M. Lapata, "Confidence modeling for neural semantic parsing," *arXiv preprint arXiv:1805.04604*, 2018.

[11] 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.

[12] A. Ogawa and T. Hori, "Error detection and accuracy estimation in automatic speech recognition using deep bidirectional recurrent neural networks," *Speech Communication*, vol. 89, pp. 70–83, 2017.

[13] A. Ragni, Q. Li, M. J. Gales, and Y. Wang, "Confidence estimation and deletion prediction using bidirectional recurrent neural networks," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 204–211.

[14] Q. Li, P. Ness, A. Ragni, and M. J. Gales, "Bi-directional lattice recurrent neural networks for confidence estimation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6755–6759.

[15] A. Kastanos, A. Ragni, and M. Gales, "Confidence estimation for black box automatic speech recognition systems using lattice recurrent neural networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6329–6333.

[16] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 427–436.

[17] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[18] A. Graves, "Practical variational inference for neural networks," in *Advances in neural information processing systems*, 2011, pp. 2348–2356.

[19] J. Heek and N. Kalchbrenner, "Bayesian inference for large scale image classification," *arXiv preprint arXiv:1908.03491*, 2019.

[20] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[21] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[22] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.

[23] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," *arXiv preprint arXiv:1612.02695*, 2016.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[25] A. Hannun, A. Lee, Q. Xu, and R. Collobert, "Sequence-to-sequence speech recognition with time-depth separable convolutions," *arXiv preprint arXiv:1904.02619*, 2019.

[26] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[27] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*, 2016, pp. 173–182.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.