AN EVALUATION OF WORD-LEVEL CONFIDENCE ESTIMATION FOR END-TO-END AUTOMATIC SPEECH RECOGNITION

Dan Oneață[1], Alexandru Caranica[1], Adriana Stan[2], Horia Cucu[1]

[1]University POLITEHNICA of Bucharest, Romania
[2]Tehnical University of Cluj-Napoca, Romania

ABSTRACT

Quantifying the confidence (or conversely the uncertainty) of a prediction is a highly desirable trait of an automatic system, as it improves the robustness and usefulness in downstream tasks. In this paper we investigate confidence estimation for end-to-end automatic speech recognition (ASR). Previous work has addressed confidence measures for lattice-based ASR, while current machine learning research mostly focuses on confidence measures for unstructured deep learning. However, as the ASR systems are increasingly being built upon deep end-to-end methods, there is little work that tries to develop confidence measures in this context. We fill this gap by providing an extensive benchmark of popular confidence methods on four well-known speech datasets. There are two challenges we overcome in adapting existing methods: working on structured data (sequences) and obtaining confidences at a coarser level than the predictions (words instead of tokens). Our results suggest that a strong baseline can be obtained by scaling the logits by a learnt temperature, followed by estimating the confidence as the negative entropy of the predictive distribution and, finally, sum pooling to aggregate at word level.

Index Terms— Confidence scoring, uncertainty estimation, automatic speech recognition, end-to-end deep learning

1. INTRODUCTION

Reasoning under uncertainty is one of the tenets of intelligence. The first step towards this goal is to endow systems with reliable uncertainty estimates of their predictions. Ideally, the larger the uncertainty the more likely the prediction is erroneous. Alternatively, one can solve the complementary problem of confidence estimation—in this case, the more confident a prediction, the more likely the output is correct.

In the context of automatic speech recognition (ASR) confidence estimation can be of crucial importance for many end-user applications, as it improves the robustness of the systems in safety-critical tasks, helps avoiding errors in human-computer dialogue systems and facilitates manual corrections in audio transcription tasks by flagging the errors. Moreover, previous research has leveraged confidence estimates for a number of downstream tasks: propagating uncertainties for automatic speech translation [1], selecting confident predictions for self-training [2], manually annotating the less confident predictions for active learning [3].

In this paper we consider confidence estimation for *end-to-end* ASR systems, also known as lattice-free speech recognition [4]. End-to-end models for ASR are gaining traction recently as their performance matches the one of classical ASR and have the additional benefits of being conceptually simple and allowing unified training [5, 6, 7]. However, there is surprisingly little work on confidence estimation for end-to-end speech recognition systems, most of the ongoing research on confidence estimation being carried on computer vision tasks (image classification or segmentation). We believe that there are two main challenges to developing confidence scoring methods for ASR systems: the structured output and the granular predictions (*e.g.*, tokens or graphemes versus words).

ASR systems are structured models (mapping sequences to sequences) as opposed to usual recognition networks (such as, image classification) whose output is a single label. The sequential nature of the output imposes a decoding step, which complicates not only the prediction but also the confidence scoring algorithm, as we need estimate the confidence in an auto-regressive context (the already predicted sequence). For this reason, we fix the predictions based on a pre-trained ASR and apply the confidence scoring methods on top of token probabilities, which are conditioned on the fixed transcript.

In order to enable open vocabulary predictions, end-to-end ASR systems usually use subword tokens to represent the output (byte-pair encoded tokens or even graphemes). However, given that the tokens lack semantics, for many downstream applications we are interested in estimating the confidence of words. To this end, we explore ways of aggregating the token-level uncertainty measures to the larger units, corresponding to words; in fact, the presented techniques can be extended to even coarser predictions, such as sentence or utterance level.

In this context, our main contributions are the following: *(i)* we adapt several state-of-the-art uncertainty estimation methods to the end-to-end ASR pipeline; *(ii)* we propose and evaluate aggregation techniques to obtain user-relevant confidence estimates (*i.e.* word-level); *(iii)* we perform a thorough evaluation on multiple speech benchmark datasets. To the best of

our knowledge, this is the first study that provides an in-depth analysis of confidence measures for end-to-end ASR.

2. RELATED WORK

In this section we review two lines of research that are related to our work.

Confidence scoring for speech recognition. Most prior work on confidence scoring for ASR targets classical systems based on the HMM-GMM paradigm. These methods first extract a set of features from the decoding lattice, acoustic or language model, and then train a classifier to predict whether the transcription is correct or not. Typical examples of features include log-likelihood of the acoustic realization, language model score, word duration, number of alternatives in the confusion network [8, 9, 10]. More recently, Swarup *et al.* have augmented the feature set with deep embeddings of the input audio and the predicted text [11], while Errattahi *et al.* have shown that the benefits of domain adaptation on the extracted features [12]. The classifiers employed by the confidence scoring methods range from conditional random fields [13, 14] and multiple layer perceptrons [15] to bidirectional recurrent neural networks [16, 17, 18].

Confidence scoring in end-to-end systems. The baseline method for confidence estimation in neural networks is to use directly the probability of the most-likely prediction [19]. However the neural networks tend to be overconfident and the probability estimates can be improved through temperature scaling [20], which typically leads to better calibration [21, 22]. The most promising direction in terms of simplicity and usefulness involves Monte Carlo estimation: Gal and Ghahramani use dropout at test time to obtain multiple predictions, which are then averaged [23], while Lakshminarayanan *et al.* average the predictions over an ensemble of networks usually trained with different initializations [24]. The latter has been show to be very reliable on challenging out-of-domain datasets [25], but coming at a high cost [22]. The literature on general confidence scoring is rich and continually evolving; the most interesting research avenues involve Bayesian averaging [26], generative models [27, 28], input perturbations [29, 30], exploiting inner activations [31, 32].

At the intersection of the two lines of research, there is the recent work of Malinin and Gales [33], which similar to us addresses the task of confidence estimation for end-to-end ASR systems. However, they are concerned with token and sentence uncertainty estimation, while we are interested in estimation at word level, and, consequently, provide more focus on the aggregation techniques. Furthermore, they employ ensembles as their primary method of confidence estimation, while we also evaluate temperature scaling and dropout methods. Dropout was previously used for obtaining confidence scores for ASR [34], but our approaches differ: in [34] multiple hypotheses are generated via dropout and then word confidences are assigned based on their frequency of appearance in the aligned hypothe-

ses; in contrast, we aggregate the posterior probabilities and not the hypotheses, which simplifies the procedure as it avoids the alignment step.

3. METHODOLOGY

This section presents the confidence estimation methodology and proposed ways of improving them. We first start with a description of the setup and the involved notation.

We consider a sequence-to-sequence model that maps an audio sequence $\mathbf{a}$ to a sequence of tokens $\mathbf{t} = (t_1, \cdots, t_T)$. The model is specified by the parameters θ, which are learned by minimizing losses such as the CTC or KL divergence on the training set. At test time the model outputs probabilities for the next token k in an auto-regressive manner $p(t_k | \hat{\mathbf{t}}_{<k}, \mathbf{a}; \theta)$ based on the already predicted tokens $\hat{\mathbf{t}}_{<k}$. These probabilities are used for performing decoding via beam search to obtain the most likely sequence of tokens. Given that the conditioned output probability is a distribution over the V tokens in the vocabulary, we denote it by a V-dimensional vector, $\mathbf{p}_k$.

The main assumption of our methodology is the availability of a probability distribution over each token. This criterion is satisfied by most end-to-end ASR architectures including the RNN transducer [35], recurrent neural aligner [36], attention-based encoder-decoder [37] and 2D LSTMs [38].

3.1. Confidence estimation

Our goal is to obtain a confidence score for each word in the output transcript of the ASR. We achieve this in two steps. First, using the posterior probabilities at each time step $\mathbf{p}_k$, we extract features to encode the confidence score of each token $s_k^{(\mathrm{t})}$. Second, we aggregate the token-level scores into word-level confidence scores $s_j^{(\mathrm{w})}$, based on the word boundaries. Next we detail these two steps; see also figure 1.

Feature extraction. To measure the confidence in a prediction at token level we use two variants:

- Log probability (log-proba) of the most probable prediction given by classifier, that is $s^{(\mathrm{t})} = \log \max \mathbf{p}$. This type of feature has been shown to yield a strong baseline for the related tasks of misclassification and out-of-distribution detection [19].

- Negative entropy (neg-entropy) computed over the vocabulary of tokens at each time stamp, that is $s^{(\mathrm{t})} = \mathbf{p}^\mathsf{T} \log \mathbf{p}$. A large entropy means a large uncertainty or, conversely, a large negative entropy implies a confident prediction.

Aggregation. To obtain word-level features from the token-level ones, we experiment with three types of aggregation functions: sum, average, minimum. Since both proposed features are negative, summing across tokens will result
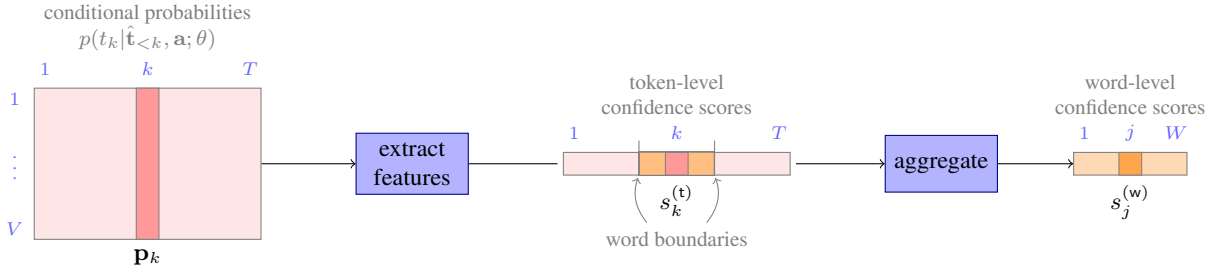
Fig. 1. Overview of the confidence scoring procedure. From an end-to-end ASR system we obtain probabilities $\mathbf{p}_k$ of the k-th token given an utterance $\mathbf{a}$ and previously predicted tokens $\hat{\mathbf{t}}_{<k}$. Based on these probabilities we extract token-level confidence scores $s^{(\mathsf{t})}$, which we then aggregate to obtain scores at word level $s^{(\mathsf{w})}$. The size of the token vocabulary is denoted by V, the number of tokens is denoted by T and the number of words by W.

in smaller values and, hence, in lower confidences; this behaviour can be desirable as longer words are more likely to be erroneous (see figure 2). Also, when we sum the log probability of the tokens, we obtain a word-level score corresponding to the log probability of the entire sequence. Taking minimum is justified by the fact that we might want a low confidence if at least one of the tokens has low confidence.

3.2. Improving the token probabilities

We propose three ways to make the token probabilities reliable: temperature scaling, dropout and ensembles of models. Our assumption is that by improving the token probabilities, we also improve the word-level scores.

Temperature scaling [20, 21] consists of dividing the logit activations (pre-softmax values) by a scalar τ (known as temperature). The value of τ ranges from zero to infinity and it controls the shape of the distribution: when $\tau \to 0$ we obtain a uniform distribution, when $\tau \to \infty$ we obtain a Dirac distribution on the most likely output. Based on τ we update token-level probabilities $\mathbf{p}$ at each time stamp k, as follows:

$$\mathbf{p}'_k = \mathrm{softmax}(\log(\mathbf{p}_k)/\tau). \tag{1}$$

We then extract features $s^{(\mathsf{t})}$ on the updated probabilities $\mathbf{p}'$, aggregate them into the word-level score $s^{(\mathsf{w})}$ and, finally, classify the word as either correct or incorrect:

$$P(\mathrm{correct}) = \sigma(\alpha \cdot s^{(\mathsf{w})} + \beta). \tag{2}$$

The variables α, β and τ are parameters and are learnt by optimizing the cross-entropy loss on a validation set. The labels are set at word level by aligning at the groundtruth text with the transcription. Note that the parameters α and β are not changing the ranking of the predictions, but allow us to learn a calibrated confidence model.

Dropout [39] is a technique that masks out random parts of the activations in a network, making the network less prone to overfitting. In [23] it has been observed that the dropout induces a probability distribution over the weights of the network

and can be consequently used for approximate Bayesian inference. We follow this idea and average the token probabilities obtained over multiple runs of dropout:

$$\mathbf{p}'_k = \frac{1}{N} \sum_n \hat{\mathbf{p}}_k \tag{3}$$

where $\hat{\mathbf{p}}$ specifies the dropout prediction. While the original work [23] employed entropy as a confidence measure, there is no reason not use other uncertainty features; we use the updated probabilities $\mathbf{p}'$ to extract both log-proba and neg-entropy features.

Ensembles [24] are based on the same idea of averaging predictions from multiple sources, but in this case the set of weights come from independently trained networks (different random seeds used in initialization and batch selection).In our case, we average the token predictions over the models:

$$\mathbf{p}'_k = \frac{1}{N} \sum_n p(t_k|\hat{\mathbf{t}}_{<k}, \mathbf{a}; \theta_n), \tag{4}$$

where $\{\theta_n\}_{n=1}^{N}$ specifies the ensemble of models. Note that we need to have the same context $\hat{\mathbf{t}}_{<k}$ for all models in the ensemble, so we use the one given by a pre-trained model.

The three presented approaches can be combined; for example, we can first update the probabilities using temperature scaling then average them using dropout. In the experimental section we will evaluate all these combinations.

4. EXPERIMENTAL SETUP

In this section, we describe the datasets used for evaluation, the ASR systems for which we build confidence estimates, and the evaluation metrics.

4.1. Datasets

We have opted for multiple publicly-available and widely-used datasets for our experimental setup.

Table 1. Size of the datasets (`test` split) used for confidence estimation evaluation.

dataset	no. utts.	duration
Libri clean	2.6K	5.4 h
Libri other	2.9K	5.3 h
TED	1.1K	2.6 h
CommonVoice	66K	72 h

LibriSpeech [40] is a corpus of approximately 1000 hours of read audiobooks derived from the LibriVox project. We use the dataset for both training the ASR and evaluating the confidence scoring: for training we use the three splits `clean100`, `clean360` and `other500`, while for development and evaluation we use the standard `clean` and `other` splits.

TED-LIUM 2 [41] consists of talks and their transcripts collected from the TED website. We use the dataset for evaluation and consequently employ only the predefined `dev` and `test` subsets.

CommonVoice [42] is a collaborative dataset of short transcripts that are read by people across the world; we use the first release of the dataset.[1] The data is used for evaluation and we defined `dev` and `test` subsets by choosing 10% random samples for each of them.

Table 1 presents the `test` size of each evaluation dataset.

4.2. ASR systems

The main ASR system is based on the pre-trained LibriSpeech model provided by the ESPNet toolkit [43]. The model implements the transformer architecture [44], takes as input 80-dimensional Mel filter banks (extracted with the Kaldi toolkit [45]) and outputs a sequence of tokens. The token vocabulary has dimension 5000 and is obtained by subword segmentation based on a unigram language model [46]. The model is trained on the 960h of the LibriSpeech dataset, which is further augmented using the SpecAugment techniques (time warping, frequency masking, time masking) [47]. For decoding we use a language model, which is also implemented as a transformer and is trained on the LibriSpeech transcriptions and other 14,500 public domain books [40]. The vocabulary of the language model consists of the same 5000 tokens as used by the ASR model.

For the ensemble experiments we re-train the ASR system using the same architecture and data, but different random seeds. We repeat the process four times obtaining four independent models. Due to computational constraints, these models were trained for a shorter number of epochs than the main system (10 versus 120), but we observed that the validation loss function curve began to flatten and that the test

performance is reasonable (5.5%±0.4 WER on Libri clean vs 2.7% obtained by the pre-trained model).

4.3. Evaluation metrics

Ideally, we want the confidence score to be correlated with the correctness of the transcription, that is, correct words should have large confidence score, while incorrect ones, low score. Following previous work [19, 31, 33], we employ metrics that are generally used for evaluating binary classifiers, but which have the discrimination threshold varied. More precisely, we measure the area under precision-recall curve (AUPR) and the area under receiver operating characteristic curve (AUROC). However, depending on what we want to focus (correctly or erroneously transcribed words) we obtain different variants: if we are interested in detecting erroneously transcribed words, we will treat the errors as the positive class; on the other hand if we are interested in the correctly transcribed words, we will treat the latter as the positive class. Hence, for AUPR we use two variants AUPRe (when errors are treated as positives) and AUPRs (when correct words are treated as positives). For AUROC the same value is obtained for either choice, so there is no need to make this distinction.

We do not evaluate calibration, since our methodology is not designed to necessarily yield a probability, but a score that is correlated with the label. The temperature scaling approach does indeed transform the score to a probability (since it learns the scaling coefficients α and β), but the same cannot be said about the other approaches (for example, negative entropy).

5. RESULTS

This section presents the experimental results. We start with an evaluation of features and their aggregations (§5.1), and then report results for the improved variants involving temperature scaling, dropout (§5.2) and ensembles (§5.3).

5.1. Features and aggregation

We evaluate the proposed uncertainty features and aggregation techniques on the four datasets described in subsection 4.1. We use the pre-trained model to obtain text predictions for all the audio files in the test split of each dataset, and then estimate the confidence based on the methodology described in subsection 3.1. Table 2 presents the results for all combinations of features and aggregations.

Comparison of features. We observe that log probability features outperform the entropy features across all settings (aggregations and datasets). The only notable exception is the CommonVoice dataset where the results are comparable.

Comparison of aggregations. Generally, the sum aggregation works better with log-proba features, while the min aggregation works better for entropy features. The sum might not be well suited for entropy features because their magnitude

Table 2. Confidence scoring results for combinations of features and aggregations on the four test splits. For all three metrics reported (AUPRe, AUPRs, AUROC) larger values are better. We indicate the word error rate of the pre-trained ASR system on each of the dataset by the figures on the right of the name.

|	feat.	agg.	Libri clean / 2.7%			Libri other / 6.0%			TED / 13.3%			CommonVoice / 28.6%		
			AUPRe	AUPRs	AUROC	AUPRe	AUPRs	AUROC	AUPRe	AUPRs	AUROC	AUPRe	AUPRs	AUROC
1	log-proba	sum	21.55	**99.21**	82.41	**29.99**	**98.10**	**81.75**	**39.97**	95.88	79.95	**48.98**	**77.71**	**64.84**
2	log-proba	min	**21.85**	99.19	**82.47**	28.64	98.06	81.66	39.74	**95.94**	**80.58**	46.79	76.74	62.67
3	log-proba	avg	20.12	99.10	80.90	26.72	97.93	80.47	38.74	95.88	80.29	44.51	75.82	60.87
4	neg-entropy	sum	17.31	**99.10**	79.97	26.37	**97.86**	79.58	34.96	95.41	77.57	**47.71**	**77.10**	**63.74**
5	neg-entropy	min	**19.94**	99.09	**80.55**	**26.75**	97.82	**79.64**	**37.55**	95.56	**79.01**	45.51	76.00	61.21
6	neg-entropy	avg	17.55	98.95	77.72	24.26	97.59	77.46	36.28	95.42	78.29	42.64	74.83	58.75
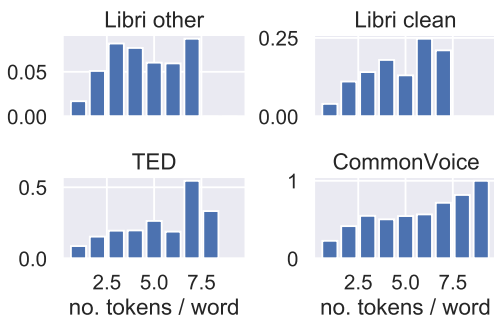
Fig. 2. Fraction of errors as a function of the word length. The fraction of errors is computed as the number of erroneously transcribed words divided by the total number of words, while the word length is measured as number of tokens.

is larger than for log-proba and the word confidence gets penalized too much by the length; but, as we will see further, this behaviour can be alleviated by temperature scaling. Averaging is generally underperforming for both features, suggesting that length-invariant measures are detrimental. Indeed, a closer look at the frequency of errors with the length size indicates that the more tokens a words has the more likely is that it is incorrect, see figure 2. Statistical tests (paired t-tests on the twelve results from each configuration at $p = 0.05$) confirm that for both features the sum and min aggregation are significantly better then avg, while the statistical test between sum and min did not reject the null hypothesis for neither feature.

Comparison across datasets. As expected, the pre-trained model performs best on in-domain data (2.7% WER on Libri clean and 6.0% on Libri other), the performance then dropping sharply as we evaluate on out-of-domain data (13.3% on TED and 28.6% on CommonVoice). In each of these settings the number of words that are correctly classified changes, going from more on the Libri splits to fewer on TED and CommonVoice. This observation explains why the performance for AUPRs drops as a function of the domain of the data, and, conversely, why the AUPRe performance improves. Unfor-

tunately, for this exact reason—the different performance of the base ASR system on the four datasets—it is impossible to compare the confidence methods across datasets, as they use a different groundtruth [22].

5.2. Temperature scaling and dropout

We benchmark the confidence scoring method after improving the token probabilities by two of the described techniques: temperature scaling and dropout. We use the pre-trained ASR system and report results only on the TED test set. The parameters for temperature scaling method are learnt on the `dev` split of the TED dataset for each setting of feature and aggregation. When temperature scaling is combined with dropout we first apply the temperature scaling (using the same temperature) and the follow with the aggregation over dropout. The dropout method averages 64 independent predictions. Table 3 presents the results for all combinations of features and aggregations and improvement techniques.

The results indicate that both proposed methods improve the results as is their combination, which gives overall the best result. We observe that log-proba features benefit more from dropout, while the neg-entropy feature yield more improvements when temperature scaling is used. Interestingly, the best results are now obtained for the neg-entropy with sum aggregation (row 16). Figure 3 shows that the dropout performance improves with the number of runs and plateaus around the chosen value of 64.

5.3. Ensembles

We present results for confidence scoring using ensembles of models and their combinations with the other improved versions (temperature scaling and dropout). For each of the retrained models from the ensemble we use the predictions of the pre-trained model to select the transcription; the retrained model is just used for confidence scoring, by extracting the confidence features described previously. The results are presented in table 4. For the rows that do not use ensemble

Table 3. Confidence scoring results on the TED test set for combinations of features, aggregations and their improved variants – temperature scaling (TS) and dropout (D). The bullet sign ● indicates whether a variant is employed. Bold results indicate the best results for the feature-aggregation combination; these results show that using both temperature scaling and dropout yields the best results.

|	feat.	agg.	TS	D	AUPRe	AUPRs	AUROC
1					39.97	95.88	79.95
2	log-proba	sum		●	41.41	96.81	82.78
3			●		40.92	96.19	81.11
4			●	●	**42.99**	**97.14**	**84.10**
5					39.74	95.94	80.58
6	log-proba	min		●	42.08	96.94	83.76
7			●		39.84	95.98	80.74
8			●	●	**42.17**	**97.00**	**83.93**
9					38.74	95.88	80.29
10	log-proba	avg		●	41.19	96.95	83.73
11			●		38.97	95.99	80.66
12			●	●	**41.32**	**97.06**	**84.08**
13					34.96	95.41	77.57
14	neg-entropy	sum		●	33.14	96.22	79.45
15			●		42.16	96.91	83.50
16			●	●	**43.59**	**97.62**	**85.51**
17					37.55	95.56	79.01
18	neg-entropy	min		●	38.75	96.53	81.98
19			●		41.23	96.87	83.50
20			●	●	**42.23**	**97.60**	**85.51**
21					36.28	95.42	78.29
22	neg-entropy	avg		●	38.01	96.51	81.85
23			●		40.22	96.53	82.48
24			●	●	**41.15**	**97.43**	**85.18**
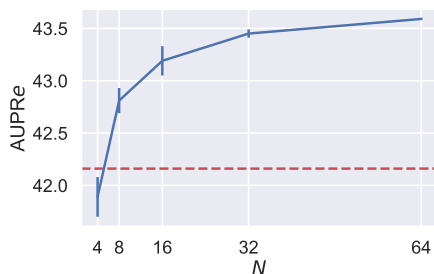
Fig. 3. AUPRe performance as a function of the number of dropout runs on the TED test set. The horizontal red line indicates the performance of the model without dropout. The model uses neg-entropy features, sum aggregation and temperature scaling.

Table 4. Confidence scoring results on the TED test set for combinations of temperature scaling (TS), dropout (D) and ensembles (E), using neg-entropy features and sum aggregation.

	TS	D	E	AUPRe	AUPRs	AUROC
1				28.58	95.30	75.79
2	●			32.00	96.32	79.47
3		●		27.49	95.51	75.67
4			●	30.89	96.26	78.89
5	●	●		31.10	96.40	79.06
6	●		●	**34.57**	**96.95**	**81.64**
7		●	●	28.94	96.26	77.93
8	●	●	●	33.00	96.84	80.82

(rows 1, 2, 3 and 5) we evaluate each of the four single models independently and report the mean performance.

The pre-trained model (table 3, row 13) has generally a better performance the retrained ones (table 4 row 1), suggesting that the predictive performance of a model can correlate with its confidence scoring performance.

Among the three improvement methods, we note that temperature scaling gives the largest performance boost on all three metrics (row 2). Surprisingly, the dropout method improves only the AUPRs performance over the baseline (row 3). On combinations of two methods, temperature scaling and ensemble complement each other and obtain better performance.

6. CONCLUSIONS

This paper presented an approach for word-level confidence scoring in end-to-end speech recognition systems. We carried a thorough ablation study on features and their aggregation on three well-known speech databases (LibriSpeech, TED-LIUM and CommonVoice) and further evaluated improved methods, which modify the token probabilities, and their combinations. Our main observation is that temperature scaling improves both uncertainty features (log-proba and neg-entropy) as well as the other two methods (dropout and ensemble). Using a pre-trained model allows replicability and enables comparison with future confidence scoring methods that will use the same ASR. We strived for simplicity by using a compact feature set (based on readily-available token posteriors); in future work we will consider augmenting these features with complementary information (*e.g.*, token duration extraction from attention).

7. ACKNOWLEDGEMENTS

8. REFERENCES

[1] Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel, "Neural lattice-to-sequence models for uncertain inputs," in *EMNLP*, 2017, pp. 1380–1389.

[2] Karel Veselỳ, Mirko Hannemann, and Lukas Burget, "Semi-supervised training of deep neural networks," in *ASRU*, 2013, pp. 267–272.

[3] Dong Yu, Balakrishnan Varadarajan, Li Deng, and Alex 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.

[4] Hossein Hadian, Hossein Sameti, Daniel Povey, and Sanjeev Khudanpur, "End-to-end speech recognition using lattice-free MMI," in *Interspeech*, 2018, pp. 12–16.

[5] Christoph Lüscher, Eugen Beck, Kazuki Irie, Markus Kitza, Wilfried Michel, Albert Zeyer, Ralf Schlüter, and Hermann Ney, "RWTH ASR systems for LibriSpeech: Hybrid vs attention," in *Interspeech*, 2019, pp. 231–235.

[6] Zoltán Tüske, Kartik Audhkhasi, and George Saon, "Advancing sequence-to-sequence based speech recognition," in *Interspeech*, 2019, pp. 3780–3784.

[7] Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, Shinji Watanabe, Takenori Yoshimura, and Wangyou Zhang, "A comparative study on transformer vs RNN in speech applications," in *ASRU*, 2019, pp. 449–456.

[8] Thomas Kemp and Thomas Schaaf, "Estimating confidence using word lattices," in *Eurospeech*, 1997.

[9] Mitch Weintraub, Francoise Beaufays, Zeév Rivlin, Yochai Konig, and Andreas Stolcke, "Neural-network based measures of confidence for word recognition," in *ICASSP*, 1997, vol. 2, pp. 887–890.

[10] Timothy J Hazen, Stephanie Seneff, and Joseph Polifroni, "Recognition confidence scoring and its use in speech understanding systems," *Computer Speech & Language*, vol. 16, no. 1, pp. 49–67, 2002.

[11] Prakhar Swarup, Roland Maas, Sri Garimella, Sri Harish Mallidi, and Björn Hoffmeister, "Improving ASR confidence scores for Alexa using acoustic and hypothesis embeddings," in *Interspeech*, 2019, pp. 2175–2179.

[12] Rahhal Errattahi, Salil Deena, Asmaa El Hannani, Hassan Ouahmane, and Thomas Hain, "Improving ASR error detection with RNNLM adaptation," in *SLT*, 2018, pp. 190–196.

[13] Mathew Stephen Seigel, *Confidence estimation for automatic speech recognition hypotheses*, Ph.D. thesis, University of Cambridge, 2013.

[14] Isaías Sánchez Cortina, Jesús Andrés-Ferrer, Alberto Sanchis, and Alfons Juan, "Speaker-adapted confidence measures for speech recognition of video lectures," *Computer Speech & Language*, vol. 37, pp. 11–23, 2016.

[15] Kaustubh Kalgaonkar, Chaojun Liu, Yifan Gong, and Kaisheng Yao, "Estimating confidence scores on ASR results using recurrent neural networks," in *ICASSP*, 2015, pp. 4999–5003.

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

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

[18] Qiujia Li, PM Ness, Anton Ragni, and Mark JF Gales, "Bi-directional lattice recurrent neural networks for confidence estimation," in *ICASSP*, 2019, pp. 6755–6759.

[19] Dan Hendrycks and Kevin Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *ICLR*, 2016.

[20] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[21] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger, "On calibration of modern neural networks," in *ICML*, 2017, pp. 1321–1330.

[22] Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov, "Pitfalls of in-domain uncertainty estimation and ensembling in deep learning," in *ICLR*, 2020.

[23] Yarin Gal and Zoubin Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *ICML*, 2016, pp. 1050–1059.

[24] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *NeurIPS*, 2017, pp. 6402–6413.

[25] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek, "Can you

trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift," in *NeurIPS*, 2019, pp. 13991–14002.

[26] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson, "A simple baseline for Bayesian uncertainty in deep learning," in *NeurIPS*, 2019, pp. 13153–13164.

[27] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan, "Do deep generative models know what they don't know?," in *ICLR*, 2018.

[28] Tong Che, Xiaofeng Liu, Site Li, Yubin Ge, Ruixiang Zhang, Caiming Xiong, and Yoshua Bengio, "Deep verifier networks: Verification of deep discriminative models with deep generative models," *arXiv preprint arXiv:1911.07421*, 2019.

[29] Shiyu Liang, Yixuan Li, and R Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *ICLR*, 2018.

[30] Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak, "On mixup training: Improved calibration and predictive uncertainty for deep neural networks," in *NeurIPS*, 2019, pp. 13888–13899.

[31] Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez, "Addressing failure prediction by learning model confidence," in *NeurIPS*, 2019, pp. 2902–2913.

[32] Tongfei Chen, Jirí Navrátil, Vijay Iyengar, and Karthikeyan Shanmugam, "Confidence scoring using whitebox meta-models with linear classifier probes," in *AISTATS*, 2019, pp. 1467–1475.

[33] Andrey Malinin and Mark Gales, "Uncertainty in structured prediction," *arXiv preprint arXiv:2002.07650*, 2020.

[34] Apoorv Vyas, Pranay Dighe, Sibo Tong, and Hervé Bourlard, "Analyzing uncertainties in speech recognition using dropout," in *ICASSP*, 2019, pp. 6730–6734.

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

[36] Hasim Sak, Matt Shannon, Kanishka Rao, and Françoise Beaufays, "Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping," in *Interspeech*, 2017, vol. 8, pp. 1298–1302.

[37] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *ICASSP*, 3 2016, pp. 4945–4949.

[38] Parnia Bahar, Albert Zeyer, Ralf Schlüter, and Hermann Ney, "On using 2d sequence-to-sequence models for speech recognition," in *ICASSP*, 2019, pp. 5671–5675.

[39] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 1 2014.

[40] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP*, April 2015, pp. 5206–5210.

[41] Anthony Rousseau, Paul Deléglise, and Yannick Estève, "Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks," in *LREC*, 2014, pp. 3935–3939.

[42] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber, "Common Voice: A massively-multilingual speech corpus," in *LREC*, 2020.

[43] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Interspeech*, 2018, pp. 2207–2211.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.

[45] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The Kaldi speech recognition toolkit," in *ASRU*, 12 2011.

[46] Taku Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *ACL*, 2018, pp. 66–75.

[47] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Interspeech*, 2019, pp. 2613–2617.