Improving ASR confidence scores for Alexa using acoustic and hypothesis embeddings

Prakhar Swarup, Roland Maas, Sri Garimella, Sri Harish Mallidi, Björn Hoffmeister

Amazon, USA

{swarupps,rmaas,srigar,mallidih}@amazon.com, bjornh@a2z.com

Abstract

In automatic speech recognition, confidence measures provide a quantitative representation used to assess the reliability of generated hypothesis text. For personal assistant devices like Alexa, speech recognition errors are inevitable due to the growing number of applications. Hence, confidence scores provide an important metric to downstream consumers to gauge the correctness of ASR hypothesis text and to subsequently initiate appropriate actions. In this work, our aim is to improve the baseline classifier based confidence model architecture by appending additional acoustic and hypothesis embeddings to the input features. Experimental results suggest that appending acoustic embeddings provides more improvements on insertion tokens as compared to appending hypothesis embeddings which improves more on substitution tokens with respect to a baseline trained on decoder features only. Appending both acoustic as well as hypothesis embeddings provides the best results with 6% relative EER reduction and 13% relative NCE increase for logistic regression classifier.

Index Terms: speech recognition, confidence measure

1. Introduction

Recent advances in the area of automatic speech recognition (ASR) have led to the development of many voice-controlled personal assistants such as Amazon Echo and Google Home. Gradually, these devices have started to support a variety of applications such as initiating a voice call, sending a voice message, controlling house appliances, playing your favorite songs and many more. The domain of these applications is ever expanding primarily due to widespread customer demand. Although we always strive for perfect recognition from ASR, due to this ever expanding domain of applications, the actual recognized utterances will invariably be erroneous. In this context, confidence scores provide extra information in addition to ASR hypothesis text to estimate the reliability of recognition. These scores are of prime importance to ASR downstream consumers since they can essentially be used to detect and recover from recognition errors. Even for mature applications where word error rate (WER) is low, a reliability measure for hypothesis words can be used to handle certain words differently. The essential idea is that it is sensible for a voice assistant to understand how confident it is about a recognition before executing the corresponding command asked by the user. If the confidence scores are low, it makes sense to initiate an additional confirmation dialog before executing a command. A few examples of applications which benefit from accurate ASR confidence scores are listed below.

- Communication and Messaging: Confidence scores can be used to impose restrictions on sensitive content such as contact names to prevent privacy breaches where the device may erroneously establish a connection/ send a message via hallucinated ASR hypothesis.
- Wakeword verification: Wakewords like 'Alexa' which are used to initiate a dialog can be passed through an additional verification service based on ASR confidence scores to make an accept/reject decision.
- Semi-supervised learning: Accurate confidence scores can be helpful in filtering out high quality machine-transcribed data which is ingested into acoustic model training.

The task of estimating a recognizer's confidence on its generated hypothesis has been a long-standing problem in ASR literature. Generally speaking, all methods proposed for computing confidence measures in speech recognition can be classified into 3 major categories[1]: (i) Model based, (ii) Posterior based, and (iii) Utterance verification. The model based approach uses predictor features generated by the ASR recognition engine with a binary classifier model to produce confidence estimates. A lot of work has been done on combining different predictor features based on pure likelihood, n-best lists, acoustic stability, hypothesis density and language model(LM) backoff [2][3][4] with binary classifiers such as linear[5], non-linear feed-forward neural networks[2] and sequence models[6]. The posterior based approach uses posterior probabilities estimated from decoding lattices or n-best lists directly as a confidence estimate[7]. However, these probabilities are believed to be over estimated due to thin lattices/n-best lists[8] which usually hampers their utility as a confidence measure. Finally, the utterance verification approach uses the log-likelihood ratio between the null hypothesis and alternative hypothesis[9] as an estimate of ASR confidence. The major problem in this approach is designing a model for the alternative hypothesis which can either be a general background model or an utterance-specific anti-model.

These approaches can be applied at various granularity levels: frame, token/word or utterance. In this work we only focus on improving the token confidence scores which are computed for every hypothesis word. To the best of our knowledge, this is the first time neural embeddings are being used for token confidence modeling. There has been prior work on using neural embeddings for other tasks such as endpoint detection[10] and device-directed utterance detection[11]. Our objective is to develop a similar embedding based framework to improve upon the baseline token confidence model. Neural embeddings are generated from 2 sources: (i) acoustics, and (ii) ASR 1-best hypothesis by training separate Long Short Term Memory(LSTM)[12] networks on acoustic features and ASR 1-best hypothesis word sequence respectively.

The paper is organized as follows: Section 2 describes our baseline token confidence model which is based on decoder features only. Section 3 describes the motivation and procedure for training acoustic and hypothesis embeddings for confidence es-

timation. Section 4 describes our experimental details including train/test data and embedding extraction details along with results. Finally, Section 5 concludes our findings and suggests future directions of work.

2. Baseline confidence model

The baseline used for this work employs a model based approach using decoder features in conjunction with a binary classifier. Decoder features are extracted from a confusion network produced by our in-house recognition engine. A confusion network is a simple linear graph used as an alternative representation of the most likely hypotheses of the decoder lattice. The arcs in the confusion network correspond to words. Along with the word ids on each arc, confusion network also contains posterior estimates of each word[13].

For token confidence estimation, every hypothesis token is represented by a 13-dimensional feature vector. Some of these features capture the number of forward links and number of nodes explored during decoding trellis construction. A large number of trellis forward links being added indicates higher confusion for a token which in turn implies a lower confidence score. Additionally, the number of confusion network choices at a token's position is also a strong indicator of token confidence. The token posterior computed over the utterance's confusion network is also used as a feature along with certain token duration-based features.

Token-level labels used for training and evaluation are binary 0/1 labels. ASR hypothesis is aligned with the ground-truth text and matching ground truth and hypothesis tokens are labeled as '1' and all others are labeled as '0'. Note that this labeling scheme will mark all substitutions and insertions as '0' while ignoring deletions since deletions essentially refer to an absence of hypothesis token. We present baseline results with a simple logistic regression based classifier using these 13 decoder features.

An important point worth mentioning here is that training data for confidence model is exclusive from training data used for the ASR acoustic model. This is because if the same training data is used for both acoustic model and confidence model then the confidence model will develop an inherent 'positive' bias due to unnaturally high token posteriors. Hence training data for confidence model is essentially the 'DEV' partition with respect to acoustic model training data.

3. Improving ASR confidence scores

3.1. Acoustic embedding

The motivation behind using acoustic information for confidence modeling is derived from improvements reported for the device-directed task in [11]. The hypothesis here is based on the well known fact that ASR recognition errors are strongly correlated to mismatched acoustic conditions such as noise, child speakers, non-native speakers etc. Our objective is to capture these acoustic characteristics in an input utterance which lead to an ASR error via neural embeddings. Decoder features described earlier for the baseline confidence model capture this information in an indirect sense. This leads us to us believe that appending an additional acoustic embedding might help the confidence model in producing better scores.

In order to generate neural acoustic embeddings, we train a LSTM network with Log filterbank energy (LFBE) based input features and frame level labels. The frame level labeling scheme uses the start and end time information of ASR hypothesis tokens along with the hypothesis-ground truth utterance alignment to generate labels as follows:

- Frame label = 1, if frame lies within boundaries of correct token
- Frame label = 0, if frame lies within boundaries of incorrect token (Substitution or Insertion)
- Frame label = 2, otherwise for optional silence, non-speech phones

For token confidence, we require an acoustic embedding for every hypothesis token. This is obtained by averaging the frame-level output of the final LSTM layer across the start and end frames of a hypothesis token. This token acoustic embedding ($a(t)$ in Figure 1) is then appended to the baseline decoder features for training and evaluation.

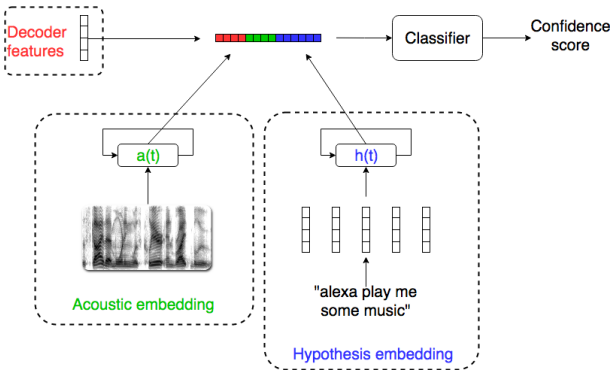


Figure 1: *Proposed token confidence model based on combination of three features types: acoustic embedding, 1-best hypothesis embedding and decoder features.*

3.2. Hypothesis embedding

Similar to acoustic embedding, we extract a token representation from ASR 1-best hypotheses. The motivation here is that certain ASR recognition errors are reflected as grammatically unstructured sentences which can be captured by a sequence trained model. To represent the word identity, using one-hot encoding would be inefficient when the vocabulary size gets large. Alternatively, word embeddings is a low-dimensional continuous space representation mapped from one-hot encoded words. Token sequence of a 1-best hypothesis is converted into vector sequence using pre-trained GloVe (Global Vectors for Word Representation) embedding vectors[14]. These word embedding vectors are used to train a LSTM network with token-level 0/1 labels identical to those described for the baseline confidence model.

For token confidence, we require a embedding for every hypothesis token. This is obtained by taking the output of the final LSTM layer for each input token. This token hypothesis embedding ($h(t)$ in Figure 1) is then appended to the baseline decoder features for training and evaluation.

4. Experiments and Results

4.1. Evaluation metrics

In an ideal scenario, confidence scores should be 1 for correct hypothesis words and 0 for incorrect hypothesis words. In order to measure how far a given set of confidence scores are from this ideal scenario a number of metrics can be used. One such metric

is Normalized Cross Entropy (NCE) which measures the relative change in cross-entropy caused when an empirical estimate of scores is replaced by the given set of confidence scores[15].

$$NCE = \frac{H(X) - H(C|X)}{H(X)} \tag{1}$$

$$H(C|X) = -\frac{1}{N}(\sum_{w \in C_h} \log(P(c|w)) + \sum_{w \in F_h} \log(1 - P(c|w)) \tag{2}$$

$$H(X) = -(p_c \log(p_c) + (1 - p_c) \log(1 - p_c)) \tag{3}$$

Here N is the total number of test tokens w in consideration, $P(c|w)$ is the output confidence score, p_c stands for the apriori probability of correctness of a token, C_h is the set of correct tokens and F_h is the set of incorrect tokens. The maximum value of NCE is 1 which corresponds to the case where hypothesized confidences match reference confidences exactly. NCE is positive if the hypothesized confidence scores are systematically better than an empirical estimate and negative otherwise.

We also report area under Receiver Operating Characteristic(ROC) curve as a metric to quantify the given set of confidence scores. An ROC curve is a plot between False Positive Rates (FPR) and True Positive Rates (TPR) obtained by varying the classification threshold (θ).

$$TPR(\theta) = \frac{TP(\theta)}{TP(\theta) + FN(\theta)} \tag{4}$$

$$FPR(\theta) = \frac{FP(\theta)}{FP(\theta) + TN(\theta)} \tag{5}$$

The ideal value of area under ROC curve is 1 and the value is 0.5 for confidence scores generated from random guessing. A particular point of interest on the ROC curve is known as the Equal Error Rate (EER) where FPR = FNR = 1 - TPR. In an ideal scenario, EER should be 0 since we would like both False Positives and False Negatives to be 0. Additionally, since confidence scores are essentially probabilities, we want to evaluate the reliability/calibration of these probabilities. Brier score is a widely used metric in this regard. For a 2 class problem, Brier score is equal to the mean square error between the ideal binary 1/0 labels and the output confidence scores. In this work we will report the Root Mean Square Error (RMSE) which is essentially the square root of Brier Score to measure the calibration of output confidence scores.

4.2. Train/Test dataset

As mentioned earlier, confidence model training dataset is typically exclusive from acoustic model training data. For this work we use a train dataset containing 520K transcribed utterances sampled from live traffic labeled as 'DEV'. This corresponds to 262 hours of speech data containing 1.73 million hypothesis tokens used for extracting decoder features and binary labels for training the confidence model. The test data used for evaluation contains 121K transcribed utterances sampled from 'TEST' traffic which corresponds to 76 hours of speech data containing 500K hypothesis tokens.

4.3. ASR model

Obtaining decoder features and labels involves decoding the training dataset through an ASR system. We use an en-US model package which consists of an Frequency LSTM[16] Hidden Markov Model (HMM) acoustic model working on Low Frame Rate (LFR) acoustic features. The acoustic features used are 256 dim Short Time Fourier Transform (STFT) features and the acoustic model consists of 2 bidirectional 16 dim LSTM layers operating along the frequency axis and 5 unidirectional 768 dim LSTM layers along the time axis. The acoustic model produces posteriors over the set of context-dependent tied HMM states which are then converted to words using our in-house lexicon and language model.

4.4. Training

Our in-house deep learning toolkit[17] is used for training in which logistic regression is formulated as a 0-hidden layer neural network. Stochastic Gradient Descent (SGD) method is used for optimizing network parameters to minimize cross-entropy loss function along with Newbob learning rate scheduling with annealing which scales the learning rate by a fixed factor if no improvement in accuracy is observed over a held-out cross validation set.

4.5. Acoustic embedding extraction

For extracting acoustic embeddings, we train a LSTM on 64 dimensional LFBE features extracted for 25 ms windows with a 10 ms hop duration. The frame level targets are generated as described in Section 3.1. The training data used for acoustic LSTM consists of 1.3M transcribed utterances containing 7.3 million hypothesis tokens sampled from 'DEV' live traffic which amounts to 1061 hours of speech data. All other training details are identical to those described earlier in Section 4.4. For evaluation, the average frame posterior of label '1' across a token is considered as token confidence score. This is compared against token binary 0/1 labels to estimate AUC, EER, NCE and RMSE metrics.

The number of layers in the LSTM network are varied to find the optimal model architecture. Empirical observations suggested that adding more layers to the LSTM improve the performance. Lowest EER is obtained by the model with 5 layers 64 cells. This result is shown in the row with features $a(t)$ in Table 1.

4.6. Hypothesis embedding extraction

For extracting hypothesis embeddings, we train a LSTM on GloVe word embeddings available for download at https://nlp.stanford.edu/projects/glove/. Word labels for training and evaluation are 1/0 binary labels depending on whether the hypothesis token matches ground truth token or not. The training data used for this LSTM is same as that used for the acoustic LSTM. All other training details are identical to those mentioned in Section 4.4.We experiment with different dimensional input word embeddings to find the optimal embedding size. The LSTM architecture used is 1 layer with cell size = word embedding size. Empirical observations suggested that increasing the word embedding size improved the performance. Lowest EER is obtained with 300 dimensional GloVe embeddings. This result is shown in the row with features $h(t)$ in Table 1.

4.7. Results

Table 1 compares the performance of baseline trained on decoder features only to a confidence model trained on different combinations of acoustic and hypothesis word embeddings.

Comparing $a(t)$ (+110% EER) and $h(t)$ (+62.7% EER) results it can be observed that while using both embeddings in-

Features	AUC	EER	NCE	RMSE
Baseline (d(t))	x	x	x	x
a(t)	-25.5 %	+110 %	-117 %	+17.1 %
h(t)	-13.4 %	+62.7 %	-53.4 %	+11.5 %
a(t)+h(t)	**-8.65 %**	**+45.9%**	**-38.6 %**	**+2.11 %**
d(t)+h(t)	+0.62 %	-2.57 %	+7.66 %	-1.47 %
d(t)+a(t)	**+1.31 %**	**-4.66 %**	**+9.66 %**	**-7.71 %**
d(t)+a(t)+h(t)	**+1.59 %**	**-6.38 %**	**+13.4%**	**-8.63 %**

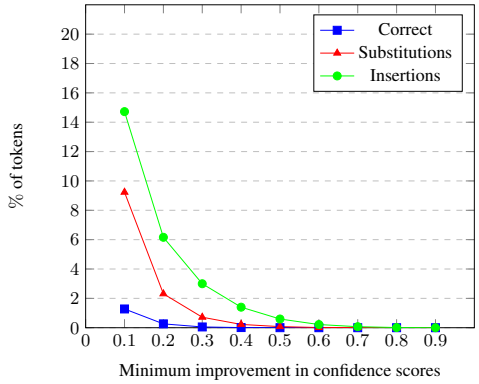
Table 1: *Token confidence model performance using different features with a logistic regression classifier with respect to baseline*

dividually significantly degrades performance with respect to baseline, the degradation is much less when using hypothesis embeddings. This might be due to the fact that the training and evaluation methodology for the acoustic LSTM are mismatched. The acoustic LSTM was trained using frame-level targets but evaluated by averaging the frame posteriors across a token. This train/eval mismatch is not present for the word LSTM since it was trained on token GloVe embeddings and evaluated on token level 1/0 labels.

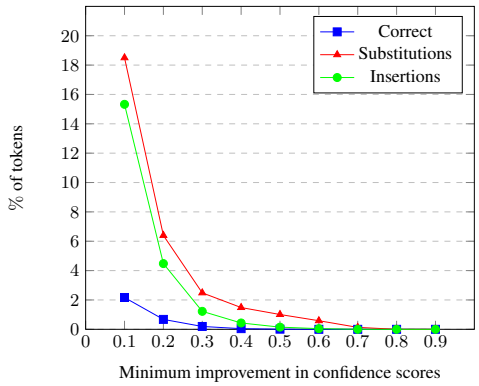
It is clear that appending either acoustic (-4.66% EER) or word embeddings (-2.57% EER) is an improvement as compared to the baseline trained on decoder features alone. This leads us to the inference that both the embeddings are complimentary in nature with respect to decoder features since the combination of features and embeddings never degrades performance. Also, both embeddings are complimentary with respect to each other because results with $a(t) + h(t)$ (+45.9% EER) are better than those with $a(t)$ or $h(t)$ alone.

Appending acoustic embeddings provides at least 2% EER and NCE improvement as compared to appending word embeddings. This suggests that acoustic embeddings capture better discriminative information with respect to correct/incorrect tokens as compared to hypothesis embeddings. In other words, discriminative information represented by hypothesis embeddings significantly overlaps that already present in baseline decoder features whereas acoustic embeddings add more complementary information for correct/incorrect token discrimination.

Further analysis of this observation is shown in Figure 2a and 2b. Here, the % of tokens having a minimum confidence improvement of x with respect to baseline is plotted for varying values of x for different kinds of tokens (Correct, Substitutions, Insertions). Improvement is measured as the simple difference in token scores between new and baseline models. Improvement refers to positive score difference for correct tokens and negative score difference for substitutions and insertions. It is clear that appending acoustic or hypothesis embeddings improves the scores of a small % of correct tokens as compared to substitutions and insertions. For a fixed minimum improvement x, appending $a(t)$ improves scores on a greater % of insertions compared to substitutions whereas appending $h(t)$ improves scores on a greater % of substitutions compared to insertions. This makes sense because most insertion errors made by our ASR model are due to acoustic conditions such as background speech and multimedia speech where we would expect acoustic embeddings to help in improving confidence scores. Also, comparing corresponding curves in Figure 2a and 2b it can be inferred that for a fixed minimum improvement x appending $h(t)$ improving scores on a greater % of substitutions as compared to $a(t)$ and appending $a(t)$ improves scores on a greater % of insertions as compared to $h(t)$.



(a) $d(t) + a(t)$ *vs.* $d(t)$



(b) $d(t) + h(t)$ *vs.* $d(t)$

Figure 2: *Comparing % of tokens with minimum confidence score improvement of x with respect to Baseline($d(t)$) for 2 different input features*

Finally, since we want to improve scores on both insertions and substitutions, appending both $a(t)$ and $h(t)$ should give us the highest improvement in performance. This is shown as the last row in Table 1 providing 6% relative EER reduction and 13% relative NCE increase with respect to baseline.

5. Conclusions

In this work, we explored appending neural acoustic and ASR 1-best hypothesis embeddings to decoder features for the task of token confidence estimation. Acoustic embedding vector is obtained by training an LSTM network on LFBEs and frame level labels based on token correctness. ASR 1-best hypothesis is obtained by first transforming the 1-best token sequence into vector sequence (using GloVe embeddings) and then training an LSTM on it. Results show that appending acoustic embeddings improves scores on insertion tokens and appending hypothesis embeddings improves scores on substitutions when compared with a baseline trained on decoder features only. Appending both acoustic as well as hypothesis embeddings provides the best results with 6% relative EER reduction and 13% relative NCE increase for logistic regression classifier. Future work in this direction would be to explore other encoder-decoder based architectures to replace vanilla LSTM networks used for extracting neural embeddings.

6. References

[1] H. Jiang, "Confidence measures for speech recognition," *Speech Communication*, vol. 45, pp. 455–470, 2005.

[2] R. San-Segundo, B. Pellom, K. Hacioglu, W. Ward, and J. M. Pardo, "Confidence measures for spoken dialogue systems," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Utah, USA, 2001.

[3] T. Schaaf and T. Kemp, "Confidence measures for spontaneous speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Munich, Germany, 1997.

[4] M. C. Benitez, A. Rubio, P. Garcia, and A. Torre, "Different confidence measures for word verification in speech recognition," *Speech Communication*, vol. 32, pp. 79–94, 2000.

[5] L. Gillick, Y. Ito, and J. Young, "A probabilistic approach to confidence estimation and evaluation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Munich, Germany, 1997.

[6] K. Kalgaonkar, C. Liu, Y. Gong, and K. Yao, "Estimating confidence scores on ASR results using recurrent neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Brisbane, Australia, 2015.

[7] F. Wessel, K. Macherey, and R. Schluter, "Using word probabilities as confidence measures," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Seattle, USA, 1998.

[8] G. Evermann and P. C. Woodland, "Large vocabulary decoding and confidence estimation using word posterior probabilities," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Istanbul, Turkey, 2000.

[9] R. C. Rose, B. H. Juang, and C. H. Lee, "A training procedure for verifying string hypotheses in continuous speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Detroit, USA, 1995.

[10] R. Maas, A. Rastrow, C. Ma, G. Lan, K. Goehner, G. Tiwari, S. Joseph, and B. Hoffmeister, "Combining acoustic embeddings and decoding features for end-of-utterance detection in real-time far-field speech recognition systems," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Alberta, Canada, 2018.

[11] S. H. Mallidi, R. Maas, K. Goehner, A. Rastrow, S. Matsoukas, and B. Hoffmeister, "Device-directed utterance detection," in *Proceedings of INTERSPEECH*, Hyderabad, India, 2018.

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

[13] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: Word error minimization and other applications of confusion networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 373–400, 2000.

[14] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.

[15] M.-H. Siu, H. Gish, and F. S. Richardson, "Improved estimation, evaluation and applications of confidence measures for speech recognition," in *Proceedings of Eurospeech*, Rhodes, Greece, 1997.

[16] T. Sainath and B. Li, "Modeling time-frequency patterns with LSTM vs. convolutional architectures for LVCSR tasks," in *Proceedings of INTERSPEECH*, San Francisco, USA, 2016.

[17] N. Strom, "Scalable distributed DNN training using commodity GPU cloud computing," in *Proceedings of INTERSPEECH*, Dresden, Germany, 2015.