Confidence-Features and Confidence-Scores for ASR applications in Arbitration and DNN Speaker Adaptation

Kshitiz Kumar, Ziad Al Bawab, Yong Zhao, Chaojun Liu, Benoit Dumoulin, Yifan Gong

Microsoft Corporation, Redmond, WA

{Kshitiz.Kumar, Ziadal, Yonzhao, Chaojunl, Bedumoul, Yifan.Gong}@microsoft.com

Abstract

Speech recognition confidence-scores quantitatively represent the correctness of decoded utterances in a [0,1] range. Confidences have primarily been used to filter out recognitions with scores below a threshold. They have also been used in other speech applications in *e.g.* arbitration, ROVER and high-quality data selection for model training etc. Confidence-scores are computed from a rich set of confidence-features in the speech recognition engine. While many speech applications consume confidence scores, we haven't seen adequate focus on directly consuming confidence-features in applications. In this work we build a thesis that additionally consuming confidence-features can provide big gains across confidence-related tasks. We demonstrate this for arbitration application, where we obtain 31% relative reduction in arbitration metric. We additionally demonstrate a novel application of confidence-scores in deep-neural-network (DNN) adaptation, where we can nearly double the relative reduction in word-error-rate (WER) for speaker adaptation on limited data.

Index Terms: Speech recognition, Confidence scores, Confidence predictors, Classifier, MLP

1. Introduction

Automatic speech recognition (ASR) has seen the strongest wave of deployment and usage across devices and services in recent years. Confidence-scores are integral to ASR, we obtain these scores from a confidence-classifier trained over a set of confidence-features to maximally discriminate between correct and incorrect recognitions. We refer [1] for an introduction to our confidence classifier framework. The Confidence-scores that lie in a [0,1] range, we desire higher scores for correct recognitions, and lower for, (a) incorrect recognitions from in-grammar (IG) and, (b) any recognition from out-of-grammar (OOG) utterances. These scores are typically evaluated for individual words as well as the utterance. Historically confidences were used for ASR-enabled devices that are always in an active (continuously) listening mode in an application-constrained grammar. There potential recognitions from side-speech, background noise etc. can trigger unexpected system response. Therefore, confidence-scores were used to contain recognitions from OOG utterances from being recognized as IG utterances. We refer [2, 3, 4, 5, 6] for a survey of confidence techniques and confidence-features. We refer [7] for a description of confidence-features obtained from decoding process, they also presented an analysis of the features with respect to reduction in cross-entropy criterion. Confidence-scores were computed from word lattices in [8], and N-best lists in [9]. We also refer [10, 11, 12, 13] for classifiers and additional features used for confidence-classification task.

Confidence-scores have also been used in other ASR applications *e.g.*, (a) arbitration where we select the best between client and service recognition results, (b) recognizer output voting error reduction (ROVER) where we perform multi-system combination [14, 15], (c) selecting high quality data for unsupervised model training, (d) key-word spotting tasks, (e) confidence-normalization [16] etc. While many of the downstream ASR applications consume confidence-scores, we have seen limited attempts on consuming confidence-features instead of confidence-score. Using confidence-features in downstream applications offers many advantages, (a) access to detailed information, (b) opportunity to retrain with confidence-features and optimize downstream task, (c) downstream application may operate over a dataset where confidence-score may not be applicable, (d) confidence-features are usually robust across languages but confidence-score may require normalization [16], (e) updating confidence-classifier does not create a dependency on updating downstream application. In this work we present our individual confidence-features, and, highlight the diverse information they encapsulate. We specifically demonstrate the richness of these features for arbitration application where we present significant gains in arbitration metric.

In addition to emphasizing the importance of confidence-features, we present a novel application of confidence-scores by embedding them in DNN speaker adaptation framework. We have established significant gains with our current speaker adaptation [17], there incorporating confidence-scores in adaptation provides additional gains. Rest of this work is organized in the following. We provide a background to our confidence-features and confidence-scores in Sec. 2. We discuss an application of these features to arbitration in Sec. 3. We present a new application of confidence-scores to further improve our baseline DNN adaptation [17] in Sec. 4. We discuss new scopes and applications of confidence-features and scores in Sec. 5. Sec. 6 concludes our study.

2. Background on Confidence-Features and Confidence-Scores

We discussed the significance of confidence-scores in Section 1 where we mentioned that confidence-classifier makes an inference on the correctness of recognition events. This is thus a binary classification problem [18] with the 2-classes in (1) correct recognitions, (2) all incorrect recognitions that includes mis-recognitions over IG utterances as well any recognition from OOG utterances. The classifier is trained from a rich set of confidence-features that we obtain from speech decoding. A few of our prominent confidence-features are:

1. acoustic-model features - we aggregate per-frame acoustic score over a word or an utterance. We also compute

scores from acoustic-arc transitions. These scores are typically normalized for duration.

2. language-model features - these include fanout and perplexity features.
3. noise and silence-model features - we compute features from noise and silence models.
4. 2nd-order features - we compute word-confidence-weighted average of acoustic features in a phrase, see [1].
5. duration features - we compute word-duration and number of words in a phrase etc.
6. senone count - count of active senones during decoding.
7. confusibility - this indicates confusibility of the best hypothesis.
8. log-spectra-derived features - we derive posterior features from speech log-Melspectra.

Our features are appropriately normalized to be robust across speech duration and intensity. We refer [1] for additional details on our confidence-classifier architecture and related features. Confidence-scores are computed from a confidence-classifier trained over confidence-features. These features are obtained from a particular collection of training data and grammar; we obtain positive tokens from successfully recognized utterances and negatives from incorrect recognitions.

Confidence-scores are optimized for classifying correct and incorrect recognitions. The optimization criterion and corresponding needs can be different for downstream speech applications that currently consume confidence-scores. In this work we motivate a use case for rich set of confidence-features. These features are typically 20-dim for a word or for an utterance, so additional memory required to store these features is minimal. The confidence-features are already computed for evaluating confidence-score so additional work required to extract confidence-features is minimal. The only incremental cost is to communicate confidence-features to the downstream application. Considering a typical speech segment of 4 secs. encoded at 8 kB/sec for a total of 32kB, confidence-features just add 80 Bytes to the communication footprint, thus less than 0.2% to speech footprint.

3. Rich Confidence-Features for Arbitration

Arbitration is an application where we select the best among multiple simultaneous ASR results. We explain our arbitration framework for personal assistant experience on smart devices in Fig. 1. We decode an utterance simultaneously for both client and service engines. Client engine is designed to work with traditional client scenarios like *call, digit dialing, text, open applications etc.*, service engine works better for rest of the speech scenarios including *voice-search, weather etc.*. By design client and service cater seamlessly to all speech scenarios and contain language-model (LM) and acoustic-model (AM) optimized for respective tasks. Though we have distinct engines for client and service speech scenarios, they work together in a unified way that is indistinct for the user as we obviously don't expect user to provide us inputs on his scenario being one of client or service. Arbitration is the key speech application that provides a unified experience by selecting the best among the client and service results. In Fig. 1 both client and service listen to speech from potentially all scenarios and produce respective recognition results under the constraint of their respective engine, AM and LM. These results are communicated to arbitration where it selects the best between the two results. Arbitration sends the results back to client where a decision unit at client will typically provide the arbitrated result to the user on their smart devices.

There can be a few scenarios where the decision unit can simply choose the client recognition *e.g.*, (a) if client confidence-scores are higher than a present threshold; client can simply choose client ASR result if it's very confident, this avoids latency incurred in hearing back from service and arbitration, (b) in absence of connection to service, then user can still use client side speech applications.

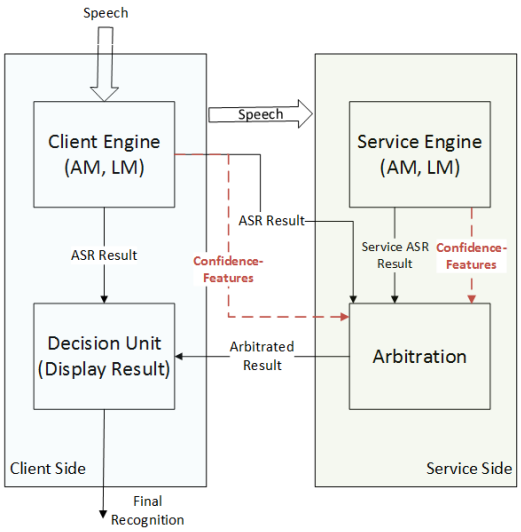

Figure 1: *Arbitration for client and service ASR results. We additionally feed confidence-features from both client and server to arbitration.*

3.1. Incorporating Confidence-Features in Arbitration

In this section, we build a thesis on consuming the rich set of confidence-features in arbitration. Our baseline arbitration is trained from a number of carefully designed features, still the confidence-features provide detailed and complementary information in noise, silence, acoustic and language-model scores, and is expected to be useful for arbitration. Confidence-scores from both client and service are currently used by arbitration, these scores provide a good gist of confidence-features but we can benefit a great deal with directly consuming confidence-features by (a) using much more gradual information in terms of 20-dim confidence-features versus a single confidence-score in arbitration, (b) confidence-scores are designed to optimize the performance of confidence-classifier which is clearly different from arbitration, so retraining with confidence-features helps, (c) arbitration and confidence-classifier may be trained over different datasets, so the information encapsulated by confidence-score may not generalize to dataset relevant for arbitration, (d) confidence-scores are language-specific as they may have been individually trained across a set of *AM, LM, languages, dataset*, in contrast, we have noted that the various inherent normalizations in confidence-features make them robust across locals, so consuming confidence-features can allow us to build an arbitra-

tion classifier from one local that can provide good performance for other unseen locals; this can be specially useful when we bootstrap arbitration for a local under limited data scenarios, (e) using confidence-score in arbitration creates a dependency for arbitration on confidences, any update to confidence-classifiers potentially requires retraining arbitration; we can completely alleviate this issue if we consume confidence-features instead of confidence-score in arbitration, this lets us independently update confidence-classifier without impacting arbitration.

We demonstrate our approach that uses the rich confidence-features for arbitration in Fig. 1. We build the infrastructure required to extract confidence-features from both client and service engine, and communicate them to arbitration, see Fig. 1. As expected we require little additional work in communicating service confidence-features to arbitration. For client, we additionally send about 30 Bytes-per-second of data, this is less than 0.2% relative increase to our payload from client to service. Depending on application and need, arbitration can be retrained and deployed with confidence-features from (a) just client, (b) just service, (c) both client and service.

3.2. Arbitration Experiments and Results

We present and analyze results with using confidence-features in arbitration. Our arbitration module was trained from over 35k speech utterances. Testing was done on over 25k utterances. We decoded these utterances against both client and service engines with their respective AM and LM, and obtained corresponding recognition results. We had ground-truth transcriptions for these utterances and created their classification targets in terms of client or service based on what provided a lower word-error rate. Our baseline arbitration features are 20-dim, that include duration, confidence-score and few semantic features etc. We additionally obtained 20-dim confidence-features from each of client and service. We followed the existing framework for training arbitration that uses a boosted-decision-tree for classification.

We demonstrate the value in client confidence-features in Fig. 2. There we plot probability-distribution-function for a few features for arbitration task. There "correct" refers to cases where client wins, and "InCorrect" refers to service wins, "Confidence-Score" indicates usual client confidence-score. We visually see that some of the features better separate the 2 classes than confidence-score.

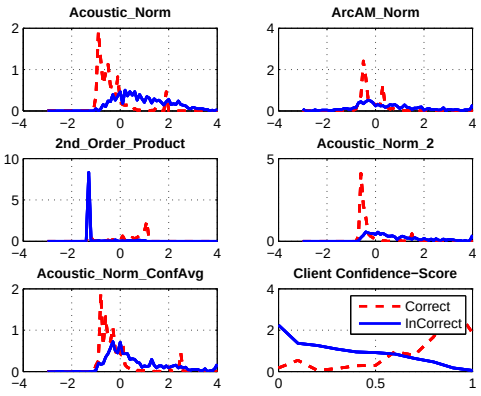

Figure 2: *Probability-distribution of representative confidence-features for arbitration.*

Table 1: Area-under-the-curve (AUC) for ROC chart

Method	AUC	relative reduction in (1-AUC) [%]
Baseline Features	0.927	-
+ Client Confidence-Features	0.946	26.0
+ Client and Service Confidence-Features	0.950	31.5

Next we provide receiver-operating-curve (ROC) for baseline and with including client confidence-features in Fig. 3, where we note a strongly better ROC curve throughout the range of curve. In this arbitration task, "False Positive" (FP) indicates incorrect wins from client and "True Positive" (TP) indicates correct wins from service. Specifically at FP of 0.1, we can improve TP from 0.81 to 0.86, for a 26% relative reduction in (1-TP). We note correct area-under-the-curve (AUC) metrics in Table 1, where including client confidence-features improved AUC from 0.927 for baseline to 0.946, additionally including server confidence-features improved AUC to 0.95 for a 31.5% relative reduction in (1-AUC) metric. Our arbitration-classifier ranks all of it's features in the order of importance. As expected confidence-features appear prominently among the top features, with 7 of the top-10 overall features being confidence-features. This further demonstrates that the proposed features outperform and add value to the baseline features.

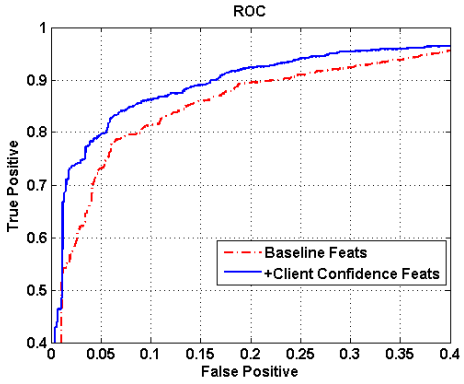

Figure 3: *Receiver operating curve (ROC) for arbitration.*

4. Confidence-Scores in DNN Speaker Adaptation Framework

In [17], we established that speaker adaption techniques can provide significant improvements to state-of-the-art DNNs-based ASR. There, we also proposed singular-value-decomposition (SVD) bottleneck adaptation, this technique involves much less adaptation parameters, while providing significant accuracy improvement. In this section we propose to embed confidence-scores in DNN adaptation. We noted that confidence-scores imply the correctness of recognition results. In the context of speaker-adaptation confidence-scores indicate a degree of match between the speaker-specific data and speaker-independent (SI) model, thus smaller confidence-scores imply weaker match between data and model, so speaker adaptation can benefit with emphasizing data with low confidence-

scores. We propose to include a weight $c(x_t)$ to standard negative cross entropy criterion [19] in below:

$$D = \frac{1}{N} \sum_{t=1}^{N} c(x_t) \sum_{s_t=1}^{S} \tilde{p}(s_t|x_t) \log p(s_t|x_t) \qquad (1)$$

There for frame t, x_t indicates input vector to DNN. Our proposed weights $c(x_t)$ depend on the confidence-scores associated with feature x_t. We group all adaptation data into 3 categories based on low, medium and high confidence-scores. We specify $c(x_t)$=2 for data associated with low confidence scores, and retain $c(x_t)$=1 for data associated with medium and high confidence-scores. In general we can specify other appropriate values for $c(x_t)$, though we found that above values of $c(x_t)$ provided consistently strong performance. Note that the baseline that doesn't embed confidence-scores is equivalent to retaining $c(x_t)$=1 for all data.

4.1. DNN adaptation experiment

We evaluated our approach on short message dictation (SMD) task and applied SVD bottleneck adaptation. The baseline SI models were trained with 300 hours VS and SMD data. Our core features are 66-dim dynamic Log-Mel-Filterbank features. We use a context window of 11 frames for a total of 726-dim input vector to DNN that has 5 hidden layers with 2,048 nodes each. The output layer has 6000 nodes. Additional details on the experimental setup can be seen in [17]. Our test set consisted of a SMD task with 6 speakers. The total number of test set words is 20,203. The baseline low-rank SI system achieves 19.93% word-error rate (WER); we also obtained confidence-scores by decoding against SI model. We varied the number of adaptation utterances from 20 (2.2 minutes) to 100 (11 minutes) for each speaker.

We note results for supervised adaptation using ground-truth transcriptions in Table 2. There for 20 utts. adaptation data we double the word-error rate relative reduction (WERR) from 1.5% to 5% when including confidence-scores in adaptation with $c(x_t)$=2 for low confidence data, and otherwise $c(x_t)$=1. We also experimented with emphasizing medium and high confidence data, and few other combinations but we found that above values of $c(x_t)$ provided consistently better results. This is understandable as here we are emphasizing DNN to learn from data that is currently a weaker match to the SI model. Medium and high confidence data is already a good match to model, so we may not learn a lot by emphasizing those data segments. Confidence-scores have also been for data selection [20] and model training in [21] but the focus was mostly on data with high confidence; our results provide a new dimension to the use of confidence-scores where we emphasize data with low confidence-scores. In Table 2, we also see strong gains for 50 and 100 utts. adaptation data. A difference of 1% WERR is significant in this task.

We report results for unsupervised adaptation in Table. 3, there we decode data against SI model and use hypothesis to align data for adaptation. There too we see 1.4-3.6% increase in WERR across different utterances. Using 20 utterances regresses baseline unsupervised adaptation, but we at least retain performance when using confidence-scores. Low confidence data from hypothesis may also indicate, (1) lower accuracy along with, (2) weaker match to SI model. However, we find merit with emphasizing low confidence data, this indicates that out of the above two factors, weaker match to SI model dominates.

Table 2: WER and WERR for Supervised adaptation. Baseline WER is 19.9%.

Nutts	Best Adaptation		+Include Confidence-Score	
	WER	WERR	WER	WERR
20	19.6	1.5	18.9	5.0
50	17.6	11.6	17.1	14.1
100	16.7	16.1	16.4	17.6

Table 3: WER and WERR for Unsupervised adaptation. Baseline WER is 19.9%.

Nutts	Best Adaptation		+Include Confidence-Score	
	WER	WERR	WER	WERR
20	20.2	-1.4	19.9	0.0
50	19.0	4.9	18.2	8.5
100	18.0	9.8	17.7	11.3

5. Discussion

We demonstrated novel applications of confidence-features and confidence-scores in this work, where we presented strong gains for arbitration and DNN speaker adaptation. We also foresee an application of confidence-features to ROVER. Confidence-score is one of the strongest features for ROVER system but these scores can be in different range across the individual systems, requiring an additional step of score normalization. We have noted that Confidence-features generalize across acoustic models and thus avoid requiring normalization. Confidence-scores are also used in model recommendation for different speakers, where we recommend one of the many AMs that may work best for particular speakers, there too we can leverage confidence-features. An approach to combine client and server ASR results while ensuring all client data remains private was recently presented in [22], confidence-features may also be applicable to that task.

6. Conclusions

Confidence-scores have been used across speech applications in ROVER, arbitration, selecting high quality data for unsupervised model training, and key-word spotting etc. Confidence-scores are computed over a rich set of confidence-features in the speech recognition engine. While a number of downstream speech applications consume confidence scores, we haven't seen adequate focus on directly consuming confidence-features in those applications. We proposed that additionally consuming confidence-features can provide huge gains for confidence-related tasks and demonstrate that with respect to arbitration, where we obtained 31% relative reduction in AUC metric. Furthermore, using confidence-features help decouple confidence-classifier and arbitration; this avoids a dependency on updating arbitration whenever we update confidence-classifier. We also demonstrate an application of confidence-scores in DNN speaker adaptation. Based on experiments we emphasize data with low confidence, this doubled WERR for DNN speaker adaptation on limited data scenarios.

7. References

[1] P.-S. Huang, K. Kumar, C. Liu, Y. Gong, and L. Deng, "Predicting speech recognition confidence using deep learning with word identity and score features," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7413–7417.

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

[3] J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing, "Confidence estimation for machine translation," in *M. Rollins (ED.), Mental Imagery*. Yale University Press, 2004.

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

[5] L. Mathan and L. Miclet, "Rejection of extraneous input in speech recognition applications, using multi-layer perceptrons and the trace of hmms," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, Apr 1991, pp. 93–96 vol.1.

[6] R. Sukkar and C.-H. Lee, "Vocabulary independent discriminative utterance verification for nonkeyword rejection in subword based speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 4, no. 6, pp. 420–429, Nov 1996.

[7] L. Chase, "Word and acoustic confidence annotation for large vocabulary speech recognition," in *in Proc. European Conference on Speech Communication Technology*, pp. 815–818.

[8] T. Kemp and T. Schaaf, "Estimating confidence using word lattices," in *Proc. of EuroSpeech*, 1997, pp. 827–830.

[9] B. Rueber, "Obtaining confidence measures from sentence probabilities." in *Eurospeech*, 1997.

[10] C. White, J. Droppo, A. Acero, and J. Odell, "Maximum entropy confidence estimation for speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, April 2007, pp. 809–812.

[11] F. Wessel, K. Macherey, and H. Ney, "A comparison of word graph and n-best list based confidence measures," in *Proc. of EuroSpeech*, 1999, pp. 315–318.

[12] F. Wessel, R. Schlter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Trans. on Speech and Audio Processing*, pp. 288–298, 2001.

[13] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, and A. Stolcke, "Neural-network based measures of confidence for word recognition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 1997, pp. 887–890.

[14] D. Hillard, B. Hoffmeister, M. Ostendorf, R. Schlüter, and H. Ney, "i rover: improving system combination with classification," in *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*. Association for Computational Linguistics, 2007, pp. 65–68.

[15] J. Fiscus, "A post-processing system to yield reduced word error rates: Recogniser output voting error reduction (rover)," in *IEEE Workshop on Automatic Speech Recognition and Understanding*. Association for Computational Linguistics, 1997, pp. 347–352.

[16] K. Kumar, C. Liu, and Y. Gong, "Normalization of asr confidence classifier scores via confidence mapping," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[17] J. Xue, J. Li, D. Yu, M. Seltzer, and Y. Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network," in *ICASSP*, 2014.

[18] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.

[19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, 1988.

[20] P. Zhang, Y. Liu, and T. Hain, "Semi-supervised dnn training in meeting recognition," in *Proceedings of IEEE Spoken Language Technology*, 2014.

[21] C. Gollan and M. Bacchiani, "Confidence scores for acoustic model adaptation," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 4289–4292.

[22] M. Georges, S. Kanthak, and D. Klakow, "Accurate client-server based speech recognition keeping personal data on the client," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2014, pp. 3271–3275.