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, 7, 8] for a survey of confidence techniques and specifically [9, 4, 10, 11, 12] for features and the classifiers used.

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) ROVER where we perform multi-system combination, (c) selecting high quality data for unsupervised model training, (d) key-word spotting tasks, (e) confidence-normalization etc. While many of the downstream ASR applications consume confidence-scores we have seen limited attempts on additionally consuming confidence-features. 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 also present a novel application of confidence-scores by embedding them in the DNN speaker adaptation framework. We have already established significant gains with our baseline speaker adaptation with limited utterances on the speaker-independent DNN model. There we embed confidence-scores into the DNN update and obtain additional gains over our best baseline adaptation.

We have seen a related framework in [13].

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 novel application of confidence-scores to further improve our current best DNN adaptation in Sec. 4. We provide new scope and application of confidence-features and scores in Sec. 5 and 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 [14] 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 may derive posterior features from speech log-spectra.

Our features are appropriately normalized to be robust to speech with different duration and intensity. We refer to [1] for additional details to our confidence-classifier architecture and related features. Though a number of speech applications consume confidence-scores, we have seen limited attempts on directly consuming the rich set of confidence-features in those applications. Confidence-scores are obtained from a confidence-classifier trained over a particular collection of training data and grammar, from which we obtain positive tokens from successfully recognized utterances and negatives from incorrect recognitions.

Thus confidence-scores are optimized for the purposes of 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 widespread use of the rich set of confidence-features. These features are typically 15-20 for a word or for an utterance, so additional memory required to store these features is minimal. These confidence-features are already computed for the purpose of confidence-score so additional work required for extracting confidence-features is almost none. Furthermore, these confidence-features will also need to be communicated along with ASR result to the downstream consumer - considering a typical speech segment of 4 secs. 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 expect to select the best among one of the multiple simultaneous ASR results. We explain our arbitration framework for Microsoft Cortana experience on smart devices in Fig. 1, where we simultaneously decode an utterance against both client and service engines, and select the best result. Client engine is designed to work with traditional client scenarios like *call, digit dialing, text, open applications etc.*, service engine works for rest of the speech scenarios including *voice-search, weather etc.*. By design the client and service cater to distinct speech scenarios and contain language-model and acoustic-model optimized for those 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. 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 2 results. Arbitration then 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 the client confidence-scores are higher than a present threshold - typically the process of getting the arbitrated ASR result back from service may incur some latency so client can simply choose client ASR result if it's very confidence about it, (b) absence of connection to service - in these cases user can still make use of client side speech applications from Microsoft or other 3^{rd}-party applications.

3.1. Arbitration Classifier and Baseline Features

Brief description of current arbitration framework, and current useful features

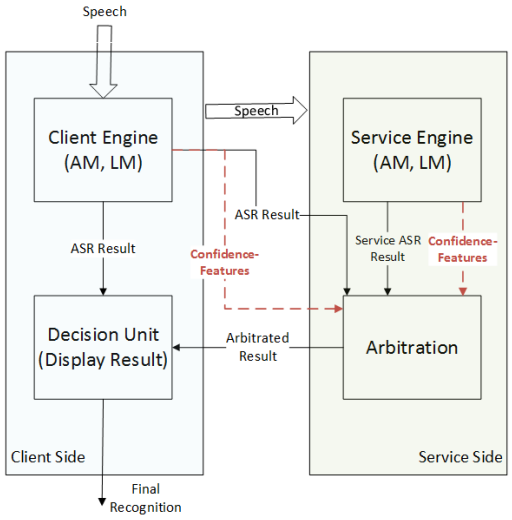

Figure 1: *Arbitration for client and service ASR resuls.*

3.2. Incorporating Confidence-Features in Arbitration

We described our arbitration framework and the baseline features in 3.1, there we mentioned a number of features that we found very relevant for arbitration. In this section, we build a thesis on consuming the rich set of confidence-features in arbitration. We first note that although our arbitration baseline features provide very useful information, the confidence-features provide detailed information in noise, silence, acoustic and language-model scores, all this is expected to useful for arbitration. We also note that although confidence-scores from both client and service is used by arbitration that are expected to provide a good gist of confidence-features, we can still benefit a great deal with directly consuming confidence-features by, (a) using much more gradual information in terms of 15-20 confidence-features vs. a single confidence-score, (b) confidence-scores are designed to optimize the performance confidence-classifier which is clearly different from arbitration, so retraining with confidence-features can help, (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 arbitration

classifier from one local that can provide good performance for other unseen locals, this can be specially useful when we may need to bootstrap arbitration from 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 alleviate this issue if we directly consume confidence-features while not using confidence-score in arbitration, this decouples arbitration and confidence-classifiers, letting 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 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.

3.3. Arbitration Experiments and Results

We present and analyze results from using confidence-features in arbitration as discussed in Sec. 3. Our arbitration 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 WER criterion. We obtained the arbitration baseline-features that included confidence-score, and additionally obtained confidence-features from both client and service. Note that ideally clients will have distinct personalized grammars with their own contact names, application names etc. For our purposes we simulated client grammar with each over 250 names in contacts and used these grammars in decoding. We followed the existing framework for training arbitration classifier where we additionally included confidence-features.

Next, we demonstrate the value in client confidence-features in Fig. 2, there we plot features distribution 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 much better separate the 2 classes than confidence-score.

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

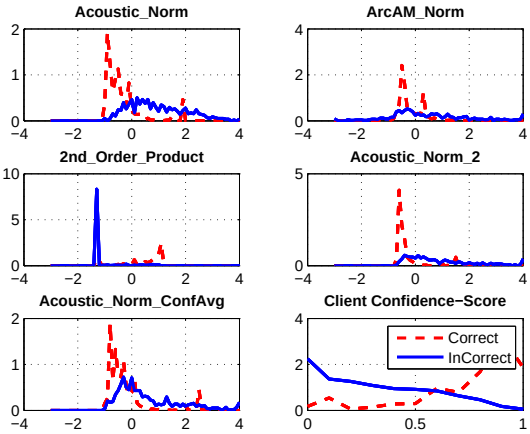

Figure 2: *Mapping to normalize confidences.*

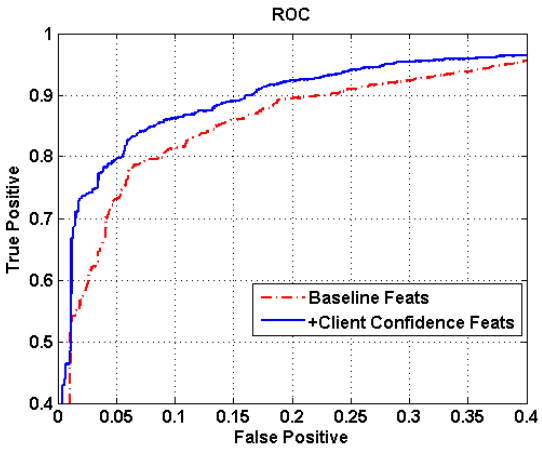

Figure 3: *Mapping to normalize confidences.*

that the proposed features outperform and add value to the baseline features.

4. Confidence-Scores in DNN Speaker Adaptation Framework

In this sec. we present a new application of confidence-scores in the DNN adaptation framework. We have huge gains in WER across many scenarios with the deployment of DNNs but we have also established that speaker adaptation of DNN models provides a significant scope for further improvements. In this section, we present our baseline speaker adaptation approach and propose to additionally include confidence-scores in speaker adaptation with limited and large adaptation data scenarios. We have noted that confidence-scores imply the correctness of recognition results. In the context of speaker-adaptation confidence-scores also indicate a degree of match between the speaker-dependent data and speaker-independent model, thus smaller confidence-scores imply weaker match between data and model. So we can leverage confidence-score in the DNN speaker adaptation optimization by disproportionately weighting optimization criterion across data based on their confidence-

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

Table 2: WER 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

scores. We create a new recipe where we collect adaptation data into 3 buckets depending on low, medium and high confidence-scores. Our goal is to change the optimization metric by including confidence-scores. Corresponding to the 3 confidence-categories we can weight the data samples from those categories according to specified values for those categories.

We know that confidence can indicate a great deal about the quality of utterances that we use in adaptation but none of the current adaptation recipes include confidence, specifically: for incorrect hypothesis, (a) low confidence data is a poor match to model and may benefit with higher weight on the data, (b) low confidence results are likely to be incorrect, so we should deemphasize these utterances, (c) high confidence data is already a good match to model, so there is less to learn from that data, (d) high confidence results are likely to be correct, so we should emphasize these utterances.

For 50 utts we can improve WERR from 11.6% to 14.1% for supervised adaptation.

Applied this recipe to training and testing on 6 speaker adaptation data on SMD task Based on experiments selected a recipe that simply duplicates the low and medium confidence buckets while retaining the high confidence bucket. Noting results for unsupervised and supervised adaptation in below

4.1. DNN adaptation experiment

Training vs test Dataset Server task Large LM Any difference of above 1% relative is significant.

Table 3: WER 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	0.2
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 in ROVER for system combination. Confidence-score is one of the strongest features for ROVER system but these confidence-scores can be in different range across the individual systems, thus they required as additional step of normalizing the confidence-scores. We have noted that Confidence-features generalize across acoustic models and thus avoid the requirement of normalization step. Confidence-scores are also used in model recommendation for identical LM, where we recommend one of the many AMs that may work best in particular scenarios. There too we can leverage confidence-features for additional gain.

6. Conclusions

Speech recognition confidence-scores quantitatively represent the correctness of decoded utterances in a [0,1] range. Confidences are primarily used to accept recognitions with scores greater than a threshold and thus contain recognitions from background noise or out-of-grammar (OOG) speech. Confidence scores have also been used in other speech applications, (a) Rover where we do multi-system combination, (b) Arbitration where we pick one among multiple simultaneous recognitions, (c) selecting high quality data for unsupervised 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 huge gains across confidence-related tasks and demonstrate that with respect to application to Arbitration, where we present 30% relative reduction in arbitration metric. We also demonstrate an application of confidence-score in deep-neural-network (DNN) speaker adaptation metric, where we can double relative reduction in word-error-rate (WER) for DNN speaker adaptation on limited data.

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] 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.

[4] 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)*, April 2007, vol. 4, pp. 809–812.

[5] 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*. 2004, Yale University Press.

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

[7] L. Mathan and Laurent 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.

[8] R.A. Sukkar and Chin-Hui 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.

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

[10] 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.

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

[12] 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.

[13] 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.

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