PREDICTING SPEECH RECOGNITION CONFIDENCE USING DEEP LEARNING WITH WORD IDENTITY AND SCORE FEATURES

Po-Sen Huang†, Kshitiz Kumar‡, Chaojun Liu‡, Yifan Gong‡, Li Deng⋆

†Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, USA
‡Microsoft Corporation, Redmond, WA, USA
⋆Microsoft Research, Redmond, WA, USA

huang146@illinois.edu, {kshitiz.kumar, chaojunl, yifan.gong, deng}@microsoft.com

ABSTRACT

Confidence classifiers for automatic speech recognition (ASR) provide a quantitative representation for the reliability of ASR decoding. In this paper, we improve the ASR confidence measure performance for an utterance using two distinct approaches: (1) to define and incorporate additional predictors in the confidence classifier including those based on the word identity and on the aggregated words, and (2) to train the confidence classifier built on deep learning architectures including the deep neural network (DNN) and the kernel deep convex network (K-DCN). Our experiments show that adding the new predictors to our multi-layer perceptron (MLP)-based baseline classifier provides 38.6% relative reduction in the correct-reject rate as our measure of the classifier performance. Further, replacing the MLP with the DNN and K-DCN provides an additional 14.5% and 47.5% in the relative performance gain, respectively.

Index Terms— Confidence measure, Word identity, Deep neural network, Kernel deep convex network

1. INTRODUCTION

Automatic speech recognition (ASR) has added significantly to a hands-free communication with devices, viz., smartphones, tablets, game consoles, etc. ASR technologies have been very successful in the past decade and have seen a rapid deployment from laboratory settings to real-life situations. Although we strive for perfect recognition from ASR, the actual decoded utterances are invariably erroneous. In this context, a confidence measure on the recognized utterance provides a quantitative representation on the reliability of an ASR decoding. This confidence measure is especially important for applications where an ASR-enabled device is always in an active listening mode in an application-constrained grammar. There it is likely that some *out-of-grammar (OOG)* utterances may still be recognized as an *in-grammar (IG)* utterance. Confidence classifiers are trained to provide a measure on the reliability of the decoded utterance in order to help reject OOG utterances. Confidence measures are also used for validating ASR decoding in presence of background noise, reverberation and other mismatched acoustic conditions. Confidence measures can be trained for word-based as well as utterance-based confidences.

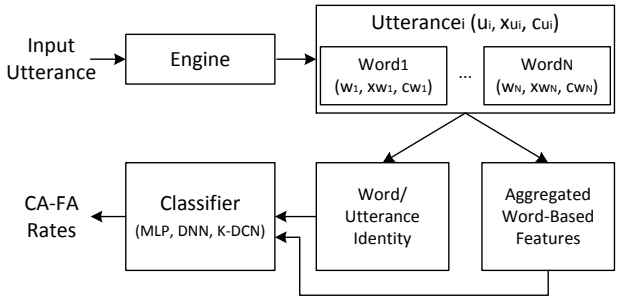

Fig. 1. Flow chart of the confidence measure system, where u_i/w_i represents i^{th} utterance/word identity, x_{u_i}/x_{w_i} represents generic features of i^{th} utterance/word, and c_{u_i}/c_{w_i} represents generic confidence measure of i^{th} utterance/word.

A number of approaches have been proposed in the area of confidence evaluation and can be categorized in 3 groups: (1) *confidence measure as combination of predictor features*: a two-class classifier is trained to determine confidences based on predictors dumped from an ASR engine [1], (2) *confidence measure as posterior probability*: confidence is estimated from posterior probability of a word or an utterance given the acoustic signal through ASR lattices [2] or N-best lists [3], (3) *confidence measure as utterance verification*: confidence is estimated from likelihood ratio between the null hypothesis and the alternative hypothesis. Refer to [4] for a survery of these techniques.

This paper evaluates a confidence measure from confidence predictors and proposes to improve confidence estimation along two distinct approaches, (1) using word/utterance identity information and aggregated word-based scores, (2)

Thanks to Dong Yu, Jinyu Li and Jason Williams for helpful discussions.

deep learning architectures including DNN and K-DCN for training classifiers. The overall flow of the confidence measure system is shown in Figure 1.

The remainder of this paper is organized as follows: Section 2 describes our confidence estimation problem formulation. Section 3 discusses our proposed features. Section 4 presents the deep learning architectures in DNN and K-DCN. The experimental setup, along with results, is described in Section 5. Section 6 concludes this study.

2. CONFIDENCE ESTIMATION PROBLEM AND APPROACH

We discussed the significance of confidence estimation in Section 1. Here we describe our specific approach for confidence estimation in terms of a binary classification problem. First, given an input sequence, a speech recognition engine produces features. Then, based on these features, a classifier needs to determine whether the input sequence is in-grammar (IG) or out-of-grammar (OOG). The input sequence can be at a word level or an utterance level. In this paper, we focus on the utterance level decision which is more relevant for end-user experience. Note that for utterance-level features, we also have their associated word-level information.

Similar to the features described in [1, 5], we use a speech recognition engine to obtain acoustic model (AM) and language model (LM)-based scores. The scores are normalized for the utterance length. Our baseline feature set (denoted as *Generic Features*) consists of 16 predictors and we use a multi-layer perceptron (MLP) for an IG vs. OOG decision making. We train an MLP for a word-level as well an utterance-level classification. The output of the MLP is the confidence score for the input utterance.

3. PROPOSED FEATURES FOR CONFIDENCE ESTIMATION

The baseline features (predictors) for the classification problem are derived from frame-level AM and LM-based scores. These frame-level scores are usually averaged across the word/utterance frame boundary, while avoiding silence regions, to obtain word/utterance-level predictors. In this work, we attempt to incorporate higher-level information in the IG vs. OOG classification problem via word-identity information. We also explicitly include the word-level confidence scores in the utterance-level confidence evaluation by variously aggregating the word-level scores.

3.1. Word/Utterance Identity Information

We can use recognized word/utterance identity information as an additional cue to extract features. Yu et. al [1] utilized word distribution information by assigning a unique ID to words with occurrence above a threshold and assigning another unique ID to words with occurrence below the threshold. Then, they represented each ID via a bag of words vector. This representation might get very long when there are a large number of high occurrence words in the corpus. Hence, we propose to grouping word/utterance identity based on their occurrences. We sort words/utterances according to their counts in the training data and divide them into K groups so that each group has the same amount of cumulative occurrence[1]. We assign a unique ID to each group as a representation of word/utterance identity - the most frequent words will be in the first group and successively words with fewer occurrence will be in latter groups. Furthermore, we assign *word identity score* by mapping word group IDs to scores from 1 to 0. The higher scores represent word group with higher occurrences.

Finally using word identities and utterance identity, we derive the following features, (1) *average word IDs*, (2) *concatenation of word IDs* (denoted as *Vec. of word IDs*)[2], and (3) *utterance ID*. For an utterance, "good luck", with utterance ID being the 5^{th} group, and words "good" and "luck" respectively being in the 2^{nd} and 1^{st} group with generic word confidences 0.2 and 0.8, the *average word IDs* is 1.5, *Vec. of word IDs* is $\{2, 1\}$, and *utterance ID* is 5.

3.2. Aggregated Word-score-based Features

As described in Section 3 the frame-level features are averaged across their word/utterance duration but this likely smears local information. In order to provide additional local information to classifiers and inspired by [6], we propose to aggregate useful statistics from word-based features. For N words in an utterance with specific word-level features being x_i and associated weights being w_i, where $i = 1, \ldots, N$, we aggregate the word-level information via the *Max*, *Min*, *Weighted Average*, *Energy*, *Magnitude*, and *Cuberoot* statistics in Table 1.

The weights w_i in Table 1 specify the relative importance of the individual word-based features. We obtain these weights from, (1) word-level generic confidence score, w_{conf}, (2) word identity score, $w_{idscore}$, and (3) summation between word-level confidence score and word identity score, $w_{conf} + w_{idscore}$, (4) multiplication operation in $w_{conf} * w_{idscore}$. Note that all of the weights are normalized for $\sum_i w_i = 1$. We realize that different features carry significantly similar information, but we also expect deep learning architectures to distill and evaluate different sources

[1]The number of groups K can be chosen from a development set. We choose $K = 10$ in this work. The unseen word in the testing set are put into the $(K + 1)^{th}$ group. Hence, there are groups 1 to $K + 1$ in total.

[2]The concatenation order follows the sorted generic confidence measurement from low to high. By cross validation, we set the concatenation dimension as three. If there are more than three words, we keep the least confident three words; on the other hand, if there are less than three words, we duplicate the least confident one [1].

Table 1. Aggregated word-based features, where word-level features are x_i and their associated weights are w_i, $i = 1, \ldots, N$.

Aggregated Method	Function
Max	$\max(x_1, \ldots, x_N)$
Min	$\min(x_1, \ldots, x_N)$
Energy	$\sum_{i=1}^{N} w_i x_i^2$
Magnitude	$(\sum_{i=1}^{N} w_i x_i^2)^{0.5}$
Cuberoot	$(\sum_{i=1}^{N} w_i x_i^2)^{0.33}$
Weighted Average	$\sum_{i=1}^{N} w_i x_i$

of information in the combined features.

4. DEEP LEARNING ARCHITECTURES

Our baseline system uses an MLP for learning the confidence decision surface. In this section, we propose to extend the MLP classifier by using deep-architectures in DNNs and K-DCN.

4.1. Deep Neural Networks

Deep neural networks (DNN) are widely being used in the state-of-the-art learning systems [7, 8, 9]. DNNs extend MLP in terms of a larger number of hidden layers. The different hidden layers can model and learn local as well as higher-order structures in the data.

4.2. Kernel Deep Convex Networks

Kernel Deep Convex Network (K-DCN) was proposed in [10, 11]. K-DCN is a kernel version of the deep convex network (DCN) [12, 13]. The architecture of DCN and K-DCN is to concatenate outputs from all previous layers and the original input data as an input to the current layer. K-DCN consists of a kernel ridge regression module, which can be expressed as:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}, \mathbf{x}_i) = \mathbf{k}(\mathbf{x})^{\mathrm{T}} \alpha \tag{1}$$

where a sample $\mathbf{x}$ is evaluated with respect to all the training samples $\{\mathbf{x}_i\}_{i=1}^{N}$, α is the regression coefficient, and vector $\mathbf{k}(\mathbf{x})$ is with element $k_n(\mathbf{x}) = k(\mathbf{x_n}, \mathbf{x})$. The regression coefficient α has a closed form solution:

$$\alpha = (\lambda \mathbf{I} + \mathbf{K})^{-1} \mathbf{Y} \tag{2}$$

where λ is the regularization parameter, $\mathbf{K} \in \mathbb{R}^{N \times N}$ is a kernel matrix with elements $K_{mn} = k(\mathbf{x}_m, \mathbf{x}_n)$, $\{\mathbf{x}_i\}_{i=1}^{N}$ are from the training set, and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^{\mathrm{T}} \in \mathbb{R}^{N \times M}$ are the M-class label vectors for training [14]. Equation (2) can be computed efficiently by using the Nyström-Woodbury approximation [11, 15, 16].

5. EXPERIMENTS

We conducted the experiments on proprietary Microsoft game-console data spread across 6 different languages in US English, British English, German, Spanish, French, and Italian. The training and test sets were significantly large and came from different usage scenarios.

5.1. Confidence Evaluation Metric

Given a test sample, the confidence classifier either accepts the sample as an IG if the output is above a pre-determined threshold or rejects it as an OOG if the output is below the threshold. Given a threshold, *correct acceptance (CA)* is solely evaluated from IG tasks, whereas *false acceptance (FA)* is evaluated from OOG tasks. We report our CA rate an average of individual CAs across FA rates of 3, 6, and 9%. We report relative gain in terms of reduction in correct reject rates (CR), where $CR = 100 - CA$.

5.2. Results for Additional Features

We examined the effectiveness of our proposed features on our baseline MLP classifier that constituted a single-hidden layer. Along with the CA rates, we also report average Kullback-Leibler (KL) divergence score from the IG and OOG sample distribution of our features. A higher KL divergence for a feature implies that the IG and OOG sample distributions are farther apart and indicates that the feature can improve confidence performance.

The experimental results are presented in Table 2. There we report our CA rate and KL divergence for the baseline features set, *Generic Features*, and then by individually adding our proposed features to the baseline set. Finally *FeatureSet* constitutes the baseline features and all of the proposed features. We observe that nearly all of our proposed features improve over the baseline CA rate as well as KL divergence scores. The concatenation of all of the features in *FeatureSet* achieves the best performance and provides 38.6% relative reduction in CR rate. We have also investigated the selection of a small subset of the overall *FeatureSet* and will report in our next work.

5.3. Results for Deep Architectures

In this section, we compare the performance between MLP, DNN, and K-DCN classifiers. We use a single hidden layer MLP classifier as the baseline. The *Generic Features* and *FeatureSet* features are selected for comparison. The experimental results are shown in Table 3 with *Avg. CA* as described in Table 2.

We note the following observations from Table 3.

- For DNN, when feature set is less complex (as in *Generic Features*), the best results are obtained from

Table 2. Comparison between different features using MLP classifier in terms of averaging CA rates at FA rate of 3, 6, and 9% and average KL divergence, where the '+ *X*' represents feature set *X* is used along with the generic features, and *FeatureSet* is the concatenation of all the features in the table.

Features	Avg. CA	KL Div.
Generic Features (baseline)	93.94	247.5
+ *Avg. word IDs*	93.66	240.1
+ *Vec. of word IDs*	94.50	217.7
+ *Utterance ID*	95.68	238.8
+ *Max*	94.44	266.7
+ *Min*	94.70	280.0
$+w_{conf}$	94.16	342.8
$+w_{idscore}$	94.39	352.5
$+w_{conf} + w_{idscore}$	94.44	382.9
$+w_{conf} * w_{conf}$	94.26	333.0
FeatureSet	96.28	389.3

Table 3. Comparison between different features and classifiers. The *Rel.* represents relative reduction in correct-reject.

	Generic Features			
Classifier	Layer	hidden units	Avg. CA	Rel.
MLP	1	5	93.94	-
DNN (w. pretraining)	1	5	94.10	2.64
DNN (w/o pretraining)	1	5	94.11	2.81
K-DCN	8	-	95.18	20.46
	FeatureSet			
Classifier	Layer	hidden units	Avg. CA	Rel.
MLP	1	5	96.28	-
DNN (w. pretraining)	3	100	96.82	14.52
DNN (w/o pretraining)	3	100	96.73	12.10
K-DCN	9	-	96.66	10.22

the same architecture as the MLP, and there is no significant improvement - this observation is similar to the results in [1]). However, when the feature set becomes more complex (as in *FeatureSet*), DNN achieves the best results with deeper layers and larger hidden units. This implies that DNN is better suited for more complex features where the inclusion of additional parameters and layers can better capture the feature discrimination.

- Experimentally, compared with DNN, K-DCN is helpful for both the generic and complex features by exploring deeper layers.

- As expected, DNN with pretraining is significantly better for *FeatureSet* than DNN without pretraining (initialized by random weights).

- For the proposed features, *FeatureSet*, DNN and K-DCN respectively achieved 47.52% and 44.88% relative improvement over the *Generic Features* with baseline classifier.

6. CONCLUSION

In many ASR application, the device is often in always-listening mode; therefore, the issue of rejecting OOG utterances is especially critical since the false recognitions can trigger undesirable response from the device. In this paper we worked on the problem of detecting and improving the confidence of ASR decoding. We proposed to include additional features in terms of word-identity and aggregated word-based scores in our baseline classification. Word-identity provides higher-level knowledge and word-score explicitly incorporates the local scores from sub-utterance units. The inclusion of additional predictors motivated us to investigate deep architectures for learning the decision boundary, for which we propose DNN and K-DCN architectures. Overall, we obtained 47.5% relative reduction in correct-reject rate with our proposed confidence predictors in the DNN framework.

7. REFERENCES

[1] D. Yu, J. Li, and L. Deng, "Calibration of confidence measures in speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2461–2473, Nov. 2011.

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

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

[5] 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.

[6] P.-S. Huang, J. Yang, M. Hasegawa-Johnson, F. Liang, and T. S Huang, "Pooling robust shift-invariant sparse representations of acoustic signals," in *Interspeech*, 2012.

[7] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504 – 507, 2006.

[8] G.E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing,*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[9] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, Nov. 2012.

[10] L. Deng, G. Tur, X. He, and D. Hakkani-Tur, "Use of kernel deep convex networks and end-to-end learning for spoken language understanding," in *IEEE Workshop on Spoken Language Technology*, 2012.

[11] P.-S. Huang, L. Deng, M. Hasegawa-Johnson, and X. He, "Random features for kernel deep convex network," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013.

[12] L. Deng and D. Yu, "Deep convex network: A scalable architecture for speech pattern classification," in *Interspeech*, 2011.

[13] L. Deng, D. Yu, and J. Platt, "Scalable stacking and learning for building deep architectures," in *Intenational Conference on Acoustics, Speech, and Signal Processing*, 2012.

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

[15] P. Drineas and M. W. Mahoney, "On the Nyström method for approximating a gram matrix for improved kernel-based learning," *Journal of Machine Learning Research*, vol. 6, pp. 2153–2175, Dec. 2005.

[16] C. Cortes, M. Mohri, and A. Talwalkar, "On the impact of kernel approximation on learning accuracy," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.