ASR ERROR DETECTION USING RECURRENT NEURAL NETWORK LANGUAGE MODEL AND COMPLEMENTARY ASR

Yik-Cheung Tam, Yun Lei, Jing Zheng and Wen Wang

Speech Technology and Research Laboratory, SRI International
333 Ravenswood Ave, Menlo Park, CA 94025, USA
{wilson,yunlei,zj,wwang}@speech.sri.com

ABSTRACT

Detecting automatic speech recognition (ASR) errors can play an important role for effective human-computer spoken dialogue system, as recognition errors can hinder accurate system understanding of user intents. Our goal is to locate errors in an utterance so that the dialogue manager can pose appropriate clarification questions to the users. We propose two approaches to improve ASR error detection: (1) using recurrent neural network language models to capture long-distance word context within and across previous utterances; (2) using a complementary ASR system. The intuition is that when two complementary ASR systems disagree on a region in an utterance, this region is most likely an error. We train a neural network predictor of errors using a variety of features. We performed experiments on both English and Iraqi Arabic ASR and observed significant improvement in error detection using the proposed methods.

Index Terms— ASR error detection, recurrent neural network language model, deep neural network acoustic model, complementary ASR

1. INTRODUCTION

In dialogue-based spoken language systems, recognition errors made by an automatic speech recognizer (ASR) can affect natural language understanding. One way to mitigate this impact is to use a dialogue manager in an attempt to correct recognition errors. Specifically, erroneous regions in an ASR hypothesis are first detected. Then the dialogue manager asks the user targeted clarification questions about the error region (e.g., the dialogue manager may request rephrasing or spelling of words) [1]. In this way, accurate ASR error detection can play a crucial role within a dialogue system.

We present two approaches to improve ASR error detection performance: (1) forward and backward recurrent neural network language models (RNNLM); and (2) combining complementary deep neural network (DNN) ASR and Gaussian mixture model (GMM) ASR. Forward and backward RNNLM provides long-distance word context within and across utterances within a dialogue session. We derive forward and backward RNNLM features based on language model scores and the distance between the current word and previous/next words from the hidden state vectors. To further leverage long-distance context, we apply incremental unsupervised RNNLM adaptation on ASR hypotheses by using previous utterances.

A natural way to minimize the problem of error detection is to reduce recognition errors. As recently reported [2], DNN ASR significantly improves recognition accuracy as compared to GMM ASR. We observe significant gains in both English and Iraqi Arabic speech recognition on the DARPA TRANSTAC corpora using DNN. However, DNN tends to produce sparse word lattices due to sharp state posteriors. Thus, word posteriors in the resulting confusion network are close to unity, and few alternative paths exist. The inability to provide word confusions makes error detection difficult, because word posterior-related features become uninformative. While GMM ASR tends to produce dense lattices, its recognition accuracy is worse than that of DNN.

In this paper, we seek both more accurate ASR hypotheses and more informative word confusions in the error regions. We propose using GMM ASR as a complementary system for error detection. First, we run DNN and GMM ASR in parallel, producing two sets of confusion networks. Using the DNN confusion network as a base, we enhance the word confusions of the DNN confusion network by adding the GMM ASR hypotheses into the DNN confusion network. Then, we use the combined confusion network together with features from the confusion slot alignment to train a neural-network word confidence predictor. Our proposed approaches not only improve error detection accuracy, but also avoid compromising recognition accuracy.

We organize the paper as follows: In Section 2, we present our ASR error detection approaches. In Section 3, we evaluate our approaches on English and Iraqi Arabic ASR using DNN and GMM acoustic models. In Sections 4 and 5, we present related work and conclusions.

2. ASR ERROR DETECTION

Given an ASR system, we decode training utterances $\{X\}$ to generate confusion networks and ASR hypotheses. We align the ASR hypotheses against the manual reference transcription to mark the errors in the ASR hypotheses. We use a binary $y = 0/1$ label on each hypothesized word to indicate error or correct. Given training examples $\{(x_i, y_i)\}$ where x_i denotes a feature vector of the i-th training example, we employ a feed-forward neural network classifier to predict the word confidence $p(y_i|x_i)$. The input features are pre-processed via global mean and variance normalization. Then, a neural network classifier is trained using backpropagation. Below are the features that we use to train the neural network classifier:

2.1. Baseline features

We extract contextual features from a confusion network at each word position j of the ASR hypothesis $W = w_1 w_2 ... w_j ... w_N$:

- log word posterior $\log p(w_j|X)$.
- log unigram probability $p(w_j)$.
- log forward 4-gram probability $p(w_j|w_{j-1}w_{j-2}w_{j-3})$.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

- log backward 4-gram probability $p(w_j|w_{j+1}w_{j+2}w_{j+3})$.
- relative word position j/N.
- log sentence length $\log N$.
- number of alternative word candidates in a confusion slot.
- log word posterior of the previous word $\log p(w_{j-1}|X)$.
- log word posterior of the next word $\log p(w_{j+1}|X)$.
- log word posterior of the previous word but one $\log p(w_{j-2}|X)$.
- log word posterior of the next word but one $\log p(w_{j+2}|X)$.
- log mean of word posteriors in a confusion slot.
- standard deviation of word posteriors in a confusion slot.
- is the previous word equal to ϵ, a null symbol corresponding to word deletion?
- is the next word equal to ϵ?
- log length of the current word w_j.

2.2. RNNLM features

A recurrent neural network language model $p(w_j|w_{j-1}, h_{j-1})$ [3] has a recursive structure that predicts a current word w_j given the previous word w_{j-1} and previous hidden state vector h_{j-1}. RNNLM can be learned using backpropagation through time to maximize the log likelihood of the training sentences.

To extract RNNLM features, we first perform a feed-forward pass on an ASR hypothesis, storing a sequence of hidden state vectors at each word position. We generate the following features at each word position j of the ASR hypothesis:

- RNNLM score $\log p(w_j|w_{j-1}, h_{j-1})$: This measures the log likelihood of the current word given the word history.
- Distance$(h_{j-1}, h_j) = \frac{1}{K}\sqrt{h_{j-1} \cdot h_j}$: Viewing RNNLM as mapping a word history into a hidden state space, this feature measures the movement from the previous to the current hidden state. K denotes the dimensionality of the hidden state vector.
- Distance$(h_j, h_{j+1}) = \frac{1}{K}\sqrt{h_j \cdot h_{j+1}}$: Similarly, this feature measures the movement from the current to the next hidden state.

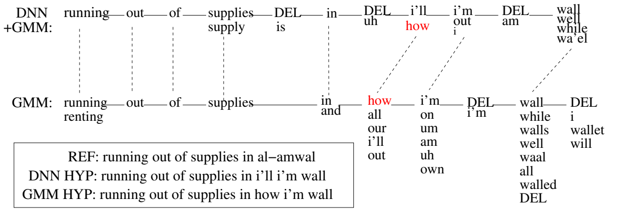
To take advantage of full sentence context, we employ a backward RNNLM $p(w_j|w_{j+1}, h_{j+1})$ trained with sentences in reverse word order. Likewise, we extract the RNNLM features described above, yielding 6 RNNLM features in total.

2.3. Incremental unsupervised RNNLM adaptation

Within a dialogue, utterances are usually correlated. We investigate incremental unsupervised RNNLM adaptation using the ASR hypothesis from the previous utterance. During adaptation, backpropagation on the previous utterance was performed with the learning rate adjusted to control the degree of adaptation.

2.4. Combining complementary ASR

System combination is an effective technique that minimizes the word error rate when multiple complementary ASR systems are available [4, 5, 6]. Extending this idea to ASR error prediction, we posit that disagreements in complementary ASR hypotheses could be a good indicator of error regions. In our experiments, DNN ASR serves as a primary system, while GMM ASR serves as complementary ASR. DNN and GMM ASR are trained on the same data but with different modeling assumptions. Therefore, we expect them to output different ASR hypotheses. Empirically, DNN tends to produce sparse confusion networks with very few paths, making ASR error detection difficult. Therefore, combining confusion networks from DNN and GMM enhances word confusions particularly in disagreement regions. Below, we list the steps for performing system combination and feature extraction:



REF: running out of supplies in al–amwal
DNN HYP: running out of supplies in i'll i'm wall
GMM HYP: running out of supplies in how i'm wall

Fig. 1. Aligning a combined confusion network and a complementary confusion network for feature extraction. The word "how" from a GMM hypothesis is inserted into the DNN confusion network.

1. Add the 1-best GMM ASR hypothesis into the DNN confusion network with a very small path posterior. Extract the baseline features described in Section 2.1 from this combined confusion network.
2. Introduce three binary GMM features: Does a left/current/right combined confusion slot contain a new word from the GMM hypothesis?
3. Align the combined confusion network with the GMM confusion network using dynamic programming. Use the slot-to-slot alignment to generate the following features:
    - expected word error $E[error(p_i(w), p_j(w)]$ where $p_i(w)$ denotes the word posterior distribution of the i-th slot in the combined confusion network and $p_j(w)$ denotes the word posterior distribution of the j-th slot in the GMM confusion network. The expectation is computed using the DNN hypothesis as the reference:

    $$E[error(p_i(w), p_j(w)] = \sum_w p_i(w) \cdot (1 - p_j(w))$$

    - the number of alternative word candidates, maximum, mean, and standard deviation of word posteriors of a GMM confusion slot described in Section 2.1. Project these features onto a combined confusion slot by using the slot-to-slot alignment. We also consider the left and right context for these features.

Figure 1 shows the confusion network alignment between the combined confusion network and the GMM confusion network. Adding the 1-best GMM ASR hypothesis with a very small path posterior enhances word confusions in the disagreement regions without changing the best path of the unmodified confusion network. In other words, the number of errors to detect remains unchanged. Thus, ASR error detection performance with different configurations is comparable. With the combined confusion network, features such as the number of alternative candidates, mean and standard deviation of word posteriors are affected by the structural change of the confusion network.

	# sentences	# error tokens
		DNN ASR
Train	951	1543 (10632)
Dev	330	495 (3756)
Test	304	496 (3646)

Table 1. Statistics of the English error prediction. Numbers in parentheses denote the number of tokens in ASR hypotheses.

	# sentences	# error tokens
		DNN ASR
Train	13378	21891 (85160)
Dev	3818	5668 (24456)
Test	3817	3817 (24418)

Table 2. Statistics of the Iraqi Arabic error prediction. Numbers in parentheses denote the number of tokens in ASR hypotheses.

3. EXPERIMENTAL SETUP

The GMM system [1] was trained discriminatively using 400 hours of the DARPA English TRANSTAC dataset; TRANSTAC was a speech-to-speech translation program targeting tactical military communication. The Iraqi Arabic system was trained with 600 hours of data. Thirteen dimensional MFCC features, augmented with first, second and third order derivatives with segmented mean and variance normalization were reduced to 40 dimensions by HLDA. The DNN system was trained on the same data as the GMM. Fifteen contextual features were concatenated and further reduced to 300 dimensions by linear discriminant analysis (i.e., the final input feature for DNN training). The DNN system had four hidden layers, each having 1200 hidden nodes, and 3000 output nodes corresponding to the clustered senone states from the decision tree of the GMM system. The DNN system was initially trained with the cross-entropy criterion for 15 epochs, followed by 1 epoch of boosted maximum mutual information training [7]. The same recipe was applied for Iraqi Arabic. On both English and Iraqi Arabic TRANSTAC test sets, we observed a 7%–20% relative reduction in word error rate when using DNN ASR compared to GMM ASR.

For language modeling, we trained trigram and 4-gram language models using modified Kneser-Ney smoothing for decoding and lattice rescoring, respectively. For the English background language models, we trained an in-domain language model with the English transcripts of the DARPA TRANSTAC Iraqi Arabic, Farsi, Pashto, Dari, and Malay collections, with a total of 16M words. We also trained a language model on a variety of out-of-domain data including Switchboard, Fisher, TDT, and Hub4 broadcast news transcripts. The in-domain and out-of-domain language models were linearly interpolated to generate the final trigram and 4-gram language models [8]. The Iraqi Arabic hierarchical class-based language models [8] were trained on the TRANSTAC Iraqi Arabic transcripts with 5M words. We used only the TRANSTAC data to train recurrent neural network language models using the RNNLM toolkit [3]. The RNNLM had 500 hidden nodes trained with backpropagation through time. We used the same data to train forward and backward 4-gram LM without pruning. The log probabilities of the language models were used as features in the baseline.

The training, development and test sets for error prediction comprised audio from the TRANSTAC test sets and sentences recorded in house with out-of-vocabulary words including named entities. These sets were not used in acoustic and language model training. Tables 1 and 2 show the number of error tokens in ASR hypotheses for English and Iraqi Arabic ASR. We aligned the reference transcriptions against the confusion networks by using the SRILM toolkit [9] to obtain target labels for training and evaluating error detection. For Iraqi Arabic, we further applied word normalization to obtain meaningful target labels due to morphological variations. The error tokens included substitution and insertion errors but not deletion errors, because deletions of reference words were unrecoverable and did not exist in the ASR hypotheses.

We employed a 3-layer neural network for word confidence prediction. The number of hidden nodes was set to six and eight for English and Iraqi Arabic, respectively, according to the error detection performance on the development set. The baseline neural network confidence predictor had sixteen input features, while the forward plus backward RNNLM extracted six features in total. We randomly picked 10% of the training data for cross validation during neural network training. For the English predictor, we applied L2 regularization over the neural network weights to prevent overfitting. Stochastic gradient descent was applied for parameter learning via minimizing the cross entropy of the output binary labels.

3.1. Performance metrics

We used the probability of miss $P(miss)$ and the false alarm (FA) rate to show the performance tradeoff. $P(miss)$ measures the rate of missing a word error. The false alarm rate measures the rate of incorrectly detecting a word error. We plotted the receiver operating characteristic (ROC) curve with $P(miss)$ against FA by varying a threshold. When a predicted word confidence was higher than the threshold, the word was classified as correct; otherwise, it was classified as incorrect. $P(miss)$ and FA are calculated as follows:

$$P(miss) = \frac{FN}{TP + FN} \quad (1)$$

$$FA = \frac{FP}{N} \quad (2)$$

where FN, TP, FN, FP and N denote the false negative, true positive, false negative, false positive, and the total number of word tokens in an ASR hypothesis, respectively. $TP + FN$ is the number of errors in the ASR hypotheses. The metrics' denominators were fixed for an ASR system. But they changed when the ASR configuration changed (for instance, when moving from GMM ASR to DNN ASR).

3.2. Error prediction results

Figure 2 shows the ROC curve using English DNN and GMM ASR. Because forward and backward 4-gram language models were used in the baseline, the additional gain from the RNNLM features may suggest that long-distance word context beyond the 4-gram was helpful. Considering the order of utterances allowed RNNLM to integrate long-distance context across previous utterances. We also compared the effect of unsupervised RNNLM adaptation using the previous ASR hypothesis. The RNNLM weights were adapted with the learning rate set to 0.05. However, the effectiveness of unsupervised adaptation was inconclusive.

Sparse DNN confusion networks tend to make error detection difficult. The word posteriors in confusion network slots were mostly close to unity, making the posterior-related features less useful for error prediction. To alleviate the impact of sparse DNN confusion networks, we used GMM ASR as a complementary system. Following the treatment described in Section 2.4, we added

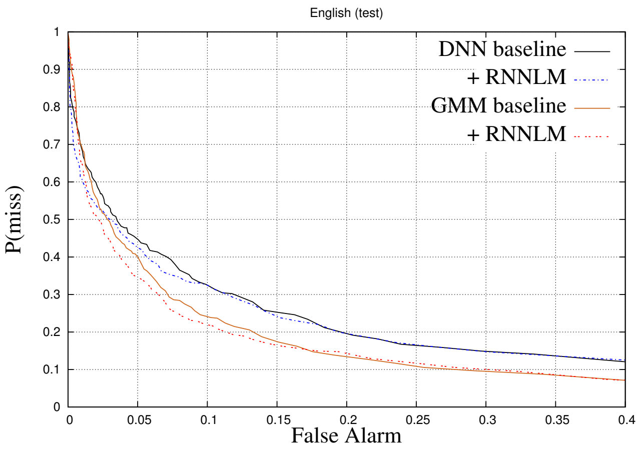


Fig. 2. ROC curve of English ASR error detection on a test set using a DNN ASR with RNNLM features.

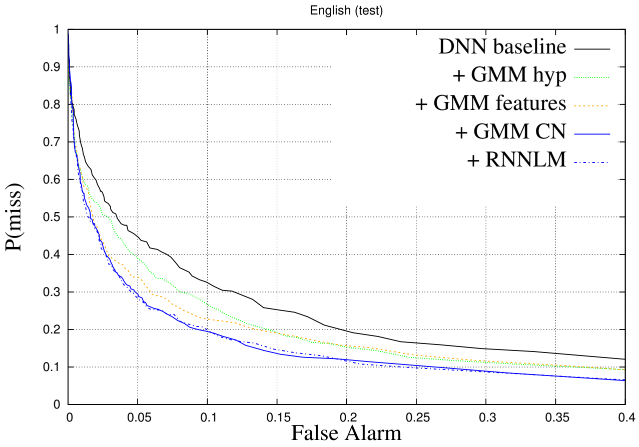


Fig. 3. ROC curve of English ASR error detection on a test set using a DNN ASR with complementary GMM ASR and RNNLM features.

the GMM ASR hypothesis into the DNN confusion network, extracted the GMM features from the combined confusion network, and extracted features from the GMM confusion network. Figure 3 showed the additive benefit using complementary GMM ASR. At a 10% false alarm rate, we achieved a 38% relative reduction in P(miss) by using GMM plus RNNLM features compared to the DNN baseline. To show the oracle performance, we added the manual reference into the DNN confusion networks. The results showed that room for improvement may exist.

Observations from Iraqi Arabic were similar, as shown in Figure 4. Combining DNN and GMM confusion networks yielded significant gain. Gain from RNNLM features and complementary GMM ASR was additive. At a 10% false alarm rate, we achieved a 30% relative reduction in P(miss) by using GMM plus RNNLM features compared to the DNN baseline. In terms of WER, Iraqi Arabic GMM ASR was inferior to DNN by a large WER margin. However, adding GMM ASR hypotheses into the DNN confusion network was still useful for ASR error prediction. This observation suggests that the disagreement regions between GMM and DNN are strong indicators of recognition errors. Although DNN was unable to produce dense lattices, adding a GMM ASR hypothesis into a DNN confusion network helped alleviate the lattice sparsity.

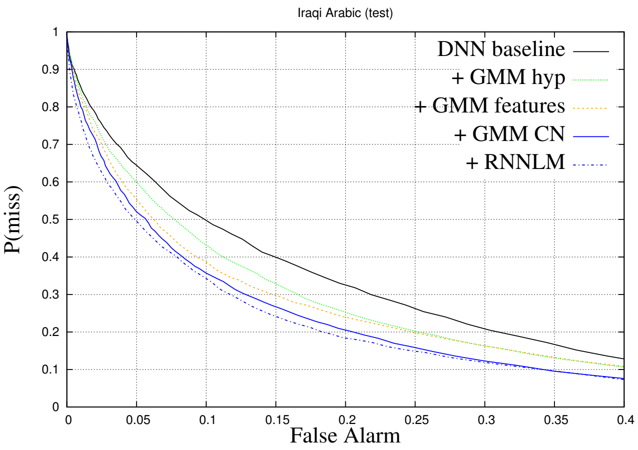


Fig. 4. ROC curve of Iraqi Arabic ASR error detection on a test set using a DNN ASR with complementary GMM ASR and RNNLM features.

4. RELATED WORK

During the previous decade, various researchers [10, 11, 12, 13, 14, 15, 16, 17] have intensively investigated confidence measures: see [18] for a survey. [19, 20] investigated features from a confusion network. [21] investigated application-dependent word distribution and rule-coverage ratio as features for confidence calibration. On a related front, [22, 23] extracted features from syntactic/dependency parsers for out-of-vocabulary detection. [24] employed word and sub-word units to train a hybrid language model to output word fragments. Research closely related to this paper exploits multiple ASR systems [25, 26, 24, 27] trained with word/phone-based acoustic models, and with multiple hybrid LMs. Our work's major differentiator resides in our use of two approaches: (1) RNNLM features to capture long-distance context within and across previous utterances and (2) combining complementary state-of-the-art DNN and GMM ASR for effective error detection. Unlike other research efforts that combine 1-best ASR hypotheses from multiple systems, we leverage DNN and GMM confusion networks that store word confusion information from multiple systems for feature extraction.

5. CONCLUSIONS

We have presented RNNLM and complementary DNN and GMM ASR for error prediction. RNNLM features capture long-distance context, while the complementary ASR helps identify ASR errors especially in disagreement regions. Results have shown significant improvement in ASR error prediction using state-of-the-art DNN ASR with the proposed approaches. In future, we plan to investigate better strategies to combine confusion networks from multiple complementary ASR systems.

6. ACKNOWLEDGEMENTS

We would like to thank Kristin Precoda, Arindam Mandal and Dimitra Vergyri for help. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-12-C-0016. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA.

7. REFERENCES

[1] N. F. Ayan, A. Mandal, M. Frandsen, J. Zheng, A. Kathol, F. Bechet, B. Favre, A. Marin, T. Kwiatkowski, M. Ostendorf, L. Zettlemoyer, P. Salletmayr, J. Hirschberg, and S. Stoyanchev, "Can you give me another word for hyperbaric?: Improving speech translation using targeted clarification questions," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vancouver, June 2013.

[2] 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, no. 6, pp. 82–97, 2012.

[3] T. Mikolov, S. Kombrink, D. Anoop, L. Burget, and J. Cernocky, "RNNLM – Recurrent neural network language modeling toolkit," in *ASRU*, 2011.

[4] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)," in *IEEE workshop on automatic speech recognition and understanding*, 1997, pp. 347–354.

[5] G. Evermann and P. C. Woodland, "Posterior probability decoding, confidence estimation and system combination," in *Proceedings of Speech Transcription workshop*, 2000.

[6] 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*, , no. 4, pp. 373–400, October 2000.

[7] G. Wang and S. Chai, "Sequential classification criteria for neural networks in automatic speech recognition," in *Proceedings of Interspeech*, 2011, pp. 441–444.

[8] M. Akbacak, H. Franco, M. Frandsen, S. Hasan, H. Jameel, A. Kathol, S. Khadivi, X. Lei, A. Mandal, S. Mansour, K. Precoda, C. Richey, D. Vergyri, W. Wang, M. Yang, and J. Zheng, "Recent advances in SRI's IraqComm(TM) Iraqi Arabic-English speech-to-speech translation system," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 2009, pp. 4809–4813.

[9] A. Stolcke, "SRILM – An extensible language modeling toolkit," in *Proceedings of International Conference on Spoken Language Processing*, Denver, Colorado, USA, September 2002, pp. 901–904.

[10] M. Siu and H. Gish, "Evaluation of word confidence for speech recognition systems," *Journal of Computer Speech and Language*, vol. 13, pp. 299–319, 1999.

[11] F. Wessel, R. Schluter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 3, pp. 288–298, 2001.

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

[13] T. Kemp and T. Schaaf, "Estimating confidence using word lattices," in *Proceedings of the European Conference on Speech Communication and Technology*, 1997, pp. 827–830.

[14] T. Schaaf and T. Kemp, "Confidence measures for spontaneous speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 875–878.

[15] L. Chase, "Word and acoustic confidence annotation for large vocabulary speech recognition," in *Proceedings of the European Conference on Speech Communication and Technology*, 1997, pp. 815–818.

[16] S. R. Young, "Word level confidence annotation using combinations of features," in *Proceedings of the European Conference on Speech Communication and Technology*, 2001.

[17] T. J. Hazen, J. Polifroni, and S. Seneff, "Recognition confidence scoring for use in speech understanding systems," *Computer Speech and Language*, vol. 16, no. 1, pp. 49–67, January 2002.

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

[19] D. Hillard and M. Ostendorf, "Compensating forward posterior estimation bias in confusion networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2006, vol. 1, pp. 1153–1156.

[20] A. Allauzen, "Error detection in confusion network," in *Proceedings of Interspeech*, 2007.

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

[22] A. Marin, T. Kwiatkowski, M. Ostendorf, and L. Zettlemoyer, "Using syntactic and confusion network structure for out-of-vocabulary word detection," in *Proceedings of Spoken Language Translation*, 2012.

[23] F. Béchét and Benoit Favre, "ASR error segment localization for spoken recovery strategy," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vancouver, June 2013.

[24] A. Rastrow, A. Sethy, and B. Ramabhadran, "A new method for OOV detection using hybrid word/fragment system," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2009, pp. 3953–3956.

[25] C. White, G. Zweig, L. Burget, P. Schwarz, and H. Hermansky, "Confidence estimation, OOV detection and language id using phone-to-word transduction and phone-level alignments," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008.

[26] L. Burget, P. Schwarz, P. Matejka, M. Hannemann, A. Rastrow, C. White, S. Khudanpur, H. Hermansky, , and J. Cernocky, "Combination of strongly and weakly constrained recognizers for reliable detection of OOVs," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008, pp. 4081–4084.

[27] L. Qin, M. Sun, and A. Rudnicky, "System combination for out-of-vocabulary word detection," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2012.