BI-DIRECTIONAL LATTICE RECURRENT NEURAL NETWORKS FOR CONFIDENCE ESTIMATION

Q. Li[†], *P. M. Ness*[†], *A. Ragni*[‡], *M. J. F. Gales*[‡]

Department of Engineering, University of Cambridge
Trumpington Street, Cambridge CB2 1PZ, UK
{ql264, pmn26, ar527, mjfg}@eng.cam.ac.uk

ABSTRACT

The standard approach to mitigate errors made by an automatic speech recognition system is to use confidence scores associated with each predicted word. In the simplest case, these scores are word posterior probabilities whilst more complex schemes utilise bi-directional recurrent neural network (BiRNN) models. A number of upstream and downstream applications, however, rely on confidence scores assigned not only to 1-best hypotheses but to all words found in confusion networks or lattices. These include but are not limited to speaker adaptation, semi-supervised training and information retrieval. Although word posteriors could be used in those applications as confidence scores, they are known to have reliability issues. To make improved confidence scores more generally available, this paper shows how BiRNNs can be extended from 1-best sequences to confusion network and lattice structures. Experiments are conducted using one of the Cambridge University submissions to the IARPA OpenKWS 2016 competition. The results show that confusion network and lattice-based BiRNNs can provide a significant improvement in confidence estimation.

***Index Terms*—** confidence estimation, bi-directional recurrent neural network, confusion network, lattice

1. INTRODUCTION

Recent years have seen an increased usage of spoken language technology in applications ranging from speech transcription [1] to personal assistants [2]. The quality of these applications heavily depends on the accuracy of the underlying automatic speech recognition (ASR) system yielding 1-best hypotheses and how well ASR errors are mitigated. The standard approach to ASR error mitigation is confidence scores [3, 4]. A low confidence can give a signal to downstream applications about the high uncertainty of the ASR in its prediction and measures can be taken to mitigate the risk of making a wrong decision. However, confidence scores can also be used in upstream applications such as speaker adaptation [5] and semi-supervised training [6, 7] to reflect uncertainty among *multiple* possible alternative hypotheses. Downstream applications, such as machine translation and information retrieval, could similarly benefit from using multiple hypotheses.

A range of confidence scores has been proposed in the literature [4]. In the simplest case, confidence scores are posterior probabilities that can be derived using approaches such as confusion networks [8, 9]. These posteriors typically significantly over-estimate confidence [9]. Therefore, a number of approaches have been proposed to rectify this problem. These range from simple piece-wise linear mappings given by decision trees [9] to more complex sequence models such as conditional random fields [10], and to neural networks [11, 12, 13]. Though improvements over posterior probabilities on 1-best hypotheses were reported, the impact of these approaches on all hypotheses available within confusion networks and lattices has not been investigated.

Extending confidence estimation to confusion network and lattice structures can be straightforward for some approaches, such as decision trees, and challenging for others, such as recurrent forms of neural networks. The previous work on encoding graph structures into neural networks [14] has mostly focused on embedding lattices into a fixed dimensional vector representation [15, 16]. This paper examines a particular example of extending a bi-directional recurrent neural network (BiRNN) [17] to confusion network and lattice structures. This requires specifying how BiRNN states are propagated in the forward and backward directions, how to merge a variable number of BiRNN states, and how target confidence values are assigned to confusion network and lattice arcs. The paper shows that the state propagation in the forward and backward directions has close links to the standard forward-backward algorithm [18]. This paper proposes several approaches for merging BiRNN states, including an attention mechanism [19]. Finally, it describes a Levenshtein algorithm for assigning targets to confusion networks and an approximate solution for lattices. Combined these make it possible to assign confidence scores to every word hypothesised by the ASR, not just from a single extracted hypothesis.

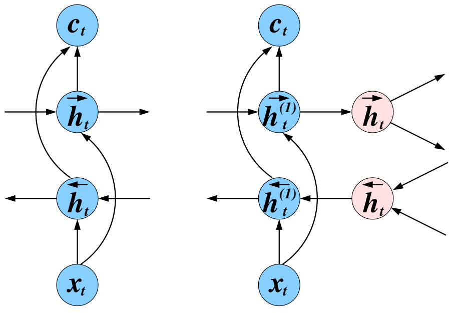
The rest of this paper is organised as follows. Section 2 describes the use of bi-directional recurrent neural networks for confidence estimation in 1-best hypotheses. Section 3 describes the extension to confusion network and lattice structures. Experimental results are presented in Section 4. The conclusions drawn from this work are given in Section 5.

2. BI-DIRECTIONAL RECURRENT NEURAL NETWORK

Fig. 1a shows the simplest form of the BiRNN [17]. Unlike its uni-directional version, the BiRNN makes use of two recurrent states, one going in the forward direction in time $\overrightarrow{\mathbf{h}}_t$ and another in the backward direction $\overleftarrow{\mathbf{h}}_t$ to model past (history) and future information respectively. The past information can be modelled by

[†] Both authors contributed equally. P. M. Ness was supported in part by the ALTA Institute, Cambridge University. [‡] Supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Air Force Research Laboratory (AFRL) contract # FA8650-17-C-9117. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, AFRL or the U.S. Government. The U.S. Government is authorised to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

978-1-5386-4658-8/18/$31.00 ©2019 IEEE



(a) sequence (b) confusion network, lattice

Fig. 1: Bi-directional neural networks for confidence estimation

$$\overrightarrow{\mathbf{h}}_t = \sigma(\mathbf{W}^{(\overrightarrow{h})}\overrightarrow{\mathbf{h}}_{t-1} + \mathbf{W}^{(x)}\mathbf{x}_t) \tag{1}$$

where $\mathbf{x}_t$ is an input feature vector at time t, $\mathbf{W}^{(x)}$ is an input matrix, $\mathbf{W}^{(\overrightarrow{h})}$ is a history matrix and σ is an element-wise non-linearity such as a sigmoid. The future information is typically modelled in the same way. At any time t the confidence c_t can be estimated by

$$c_t = \sigma(\mathbf{w}^{(c)^{\mathsf{T}}}\mathbf{h}_t + b^{(c)}) \tag{2}$$

where $\mathbf{w}^c$ and $b^{(b)}$ are a parameter vector and a bias, σ is any non-linearity that maps confidence score into the range $[0, 1]$ and $\mathbf{h}_t$ is a context vector that combines the past and future information.

$$\mathbf{h}_t = \begin{bmatrix}\overrightarrow{\mathbf{h}}_t & \overleftarrow{\mathbf{h}}_t\end{bmatrix}^{\mathsf{T}} \tag{3}$$

The input features $\mathbf{x}_t$ play a fundamental role in the model's ability to assign accurate confidence scores. Numerous hand-crafted features have been proposed [20, 21, 22, 23]. In the simplest case, duration and word posterior probability can be used as input features. More complex features may include embeddings [24], acoustic and language model scores and other information. The BiRNN can be trained by minimising the binary cross-entropy

$$H(\mathbf{c}, \mathbf{c}^*; \boldsymbol{\theta}) = -\frac{1}{T}\sum_{t=1}^{T}\left\{c^*_t \log(c_t) + (1 - c^*_t)\log(1 - c_t)\right\} \tag{4}$$

where c_t is a predicted confidence score for time slot t and c^*_t is the associated reference value. The reference values can be obtained by aligning the 1-best ASR output and reference text using the Levenshtein algorithm. Note that deletion errors cannot be handled under this framework and need to be treated separately [23, 13]. This form of BiRNN has been examined for confidence estimation in [12, 13]

The perfect confidence estimator would assign scores of one and zero to correctly and incorrectly hypothesised words respectively. In order to measure the accuracy of confidence predictions, a range of metrics have been proposed. Among these, normalised cross-entropy (NCE) is the most frequently used [25]. NCE measures the relative change in the binary cross-entropy when the empirical estimate of ASR correctness, P_c, is replaced by predicted confidences $\mathbf{c} = c_1, \ldots, c_T$. Using the definition of binary cross-entropy in Eqn. 4, NCE can be expressed as

$$\text{NCE}(\mathbf{c}, \mathbf{c}^*) = \frac{H(P_c \cdot \mathbf{1}, \mathbf{c}^*) - H(\mathbf{c}, \mathbf{c}^*)}{H(P_c \cdot \mathbf{1}, \mathbf{c}^*)} \tag{5}$$

where $\mathbf{1}$ is a length T vector of ones, and the empirical estimate of ASR correctness is given by

$$P_c = \frac{1}{T}\sum_{t=1}^{T} c^*_t \tag{6}$$

When hypothesised confidence scores $\mathbf{c}$ are systematically better than the estimate of ASR correctness P_c, NCE is positive. In the limit of perfect confidence scores, NCE approaches one.

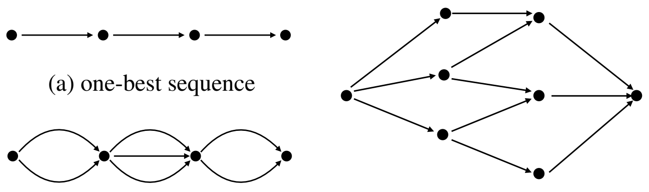
NCE alone is not always the most optimal metric for evaluating confidence estimators. This is because the theoretical limit of correct words being assigned a score of one and incorrect words a score of zero is not necessary for perfect operation of an upstream or downstream application. Often it is sufficient that the rank ordering of the predictions is such that all incorrect words fall below a certain threshold, and all correct words above. This is the case, for instance, in various information retrieval tasks [26, 27]. A more suitable metric in such cases could be an area under a curve (AUC)-type metric. For balanced data the chosen curve is often the receiver operation characteristics (ROC). Whereas for imbalanced data, as is the case in this work, the precision-recall (PR) curve is normally used [28]. The PR curve is obtained by plotting precision versus recall

$$\text{Precision}(\theta) = \frac{\text{TP}(\theta)}{\text{TP}(\theta) + \text{FP}(\theta)}, \ \text{Recall}(\theta) = \frac{\text{TP}(\theta)}{\text{TP}(\theta) + \text{FN}(\theta)} \tag{7}$$

for a range of thresholds θ, where TP are true positives, FP and FN are false positives and negatives. When evaluating performance on lattices and confusion networks, these metrics are computed across all arcs in the network.

3. CONFUSION NETWORK AND LATTICE EXTENSIONS

A number of important downstream and upstream applications rely on accurate confidence scores in graph-like structures, such as confusion networks (CN) in Fig. 2b and lattices in Fig. 2c, where arcs connected by nodes represent hypothesised words. This section describes an extension of BiRNNs to CNs and lattices.



(a) one-best sequence

(b) confusion network (c) word lattice

Fig. 2: Standard ASR outputs

Fig. 2b shows that compared to 1-best sequences in Fig. 2a, each node in a CN may have multiple incoming arcs. Thus, a decision needs to be made on how to optimally propagate information to the outgoing arcs. Furthermore, any such approach would need to handle a variable number of incoming arcs. One popular approach [16, 15] is to use a weighted combination

$$\overrightarrow{\mathbf{h}}_t = \sum_i \alpha^{(i)}_t \overrightarrow{\mathbf{h}}^{(i)}_t \tag{8}$$

where $\overrightarrow{\mathbf{h}}^{(i)}_t$ represents the history information associated with the i^{th} arc of the t^{th} CN bin and $\alpha^{(i)}_t$ is the associated weight. A number of approaches can be used to set these weights. One simple approach is to set weights of all arcs other than the one with the highest posterior to zero. This yields a model that for 1-best hypotheses has no advantage over BiRNNs in Section 2. Other simple approaches include average or normalised confidence score $\alpha_t^{(i)} = c_t^{(i)} / \sum_j c_t^{(j)}$ where $c_t^{(i)}$ is a word posterior probability, possibly mapped by decision trees. A more complex approach is an attention mechanism

$$\alpha_t^{(i)} = \frac{\exp(z_t^{(i)})}{\sum_j \exp(z_t^{(j)})}, \text{ where } z_t^{(i)} = \sigma\left(\mathbf{w}^{(a)\mathsf{T}} \overrightarrow{\mathbf{k}}_t^{(i)} + b^{(a)}\right) \quad (9)$$

where $\mathbf{w}^{(a)}$ and $b^{(a)}$ are attention parameters, $\overrightarrow{\mathbf{k}}_t^{(i)}$ is a key. The choice of the key is important as it helps the attention mechanism decide which information should be propagated. It is not obvious a priori what the key should contain. One option is to include arc history information as well as some basic confidence score statistics

$$\overrightarrow{\mathbf{k}}_t^{(i)} = \begin{bmatrix} \overrightarrow{\mathbf{h}}_t^{(i)\mathsf{T}} & c_t^{(i)} & \mu_t & \sigma_t \end{bmatrix}^{\mathsf{T}} \quad (10)$$

where μ_t and σ_t are the mean and standard deviation computed over $c_t^{(i)}$ at time t. At the next $(t+1)^{\text{th}}$ CN bin the forward information associated with the i^{th} arc is updated by

$$\overrightarrow{\mathbf{h}}_{t+1}^{(i)} = \sigma(\mathbf{W}^{(\overrightarrow{h})} \overrightarrow{\mathbf{h}}_t + \mathbf{W}^{(x)} \mathbf{x}_{t+1}^{(i)}) \quad (11)$$

The confidence score for each CN arc is computed by

$$c_t^{(i)} = \sigma(\mathbf{w}^{(c)\mathsf{T}} \mathbf{h}_t^{(i)} + b^{(c)}) \quad (12)$$

where $\mathbf{h}_t^{(i)}$ is an arc context vector

$$\mathbf{h}_t^{(i)} = \begin{bmatrix} \overrightarrow{\mathbf{h}}_t^{(i)} & \overleftarrow{\mathbf{h}}_t^{(i)} \end{bmatrix} \quad (13)$$

A summary of dependencies in this model is shown in Fig. 1b for a CN with 1 arc in the t^{th} bin and 2 arcs in the $(t+1)^{\text{th}}$ bin.

As illustrated in Fig. 2c, each node in a lattice marks a timestamp in an utterance and each arc represents a hypothesised word with its corresponding acoustic and language model scores. Although lattices do not normally obey a linear graph structure, if they are traversed in the topological order, no changes are required to compute confidences over lattice structures. The way the information is propagated in these graph structures is similar to the forward-backward algorithm [18]. There, the forward probability at time t is

$$\overrightarrow{h}_{t+1}^{(i)} = \overrightarrow{h}_t x_{t+1}^{(i)}, \text{ where } \overrightarrow{h}_t = \sum_j \alpha_{i,j} \overrightarrow{h}_t^{(j)} \quad (14)$$

Compared to equations Eqn. 8 and Eqn. 11, the forward recursion employs a different way to combine features $x_{t+1}^{(i)}$ and node states $\overrightarrow{h}_t$, and maintains stationary weights, *i.e.* the transition probabilities $\alpha_{i,j}$, for combining arc states $\overrightarrow{h}_t^{(j)}$. In addition, each $\overrightarrow{h}_t^{(i)}$ has a probabilistic meaning which the vector $\overrightarrow{\mathbf{h}}_t^{(i)}$ does not. Furthermore, unlike in the standard algorithm, the past information at the final node is not constrained to be equal to the future information at the initial node.

In order to train these models, each arc of a CN or lattice needs to be assigned an appropriate reference confidence value. For aligning a reference word sequence to another sequence, the Levenshtein algorithm can be used. The ROVER method has been used to iteratively align word sequences to a pivot reference sequence to construct CNs [29]. This approach can be extended to confusion network combination (CNC), which allows the merging of two CNs [30]. The reduced CNC alignment scheme proposed here uses a reference one-best sequence rather than a CN as the pivot, in order to tag CN arcs against a reference sequence. A soft loss of aligning reference word ω_τ with the t^{th} CN bin is used

$$\ell_t(\omega_\tau) = 1 - P_t(\omega_\tau) \quad (15)$$

where $P_t(\omega)$ is a word posterior probability distribution associated with the CN bin at time t. The optimal alignment is then found by minimising the above loss.

The extension of the Levenshtein algorithm to lattices, though possible, is computationally expensive [31]. Therefore approximate schemes are normally used [32]. Common to those schemes is the use of information about the overlap of lattice arcs and time-aligned reference words to compute the loss

$$o_{t,\tau} = \max\left\{0, \frac{|\min\{e_\tau^*, e_t\}| - |\max\{s_\tau^*, s_t\}|}{|\max\{e_\tau^*, e_t\}| - |\min\{s_\tau^*, s_t\}|}\right\} \quad (16)$$

where $\{s_t, e_t\}$ and $\{s_\tau^*, e_\tau^*\}$ are start and end times of lattice arcs and time-aligned words respectively. In order to yield "hard" 0 or 1 loss a threshold can be set either on the loss or the amount of overlap.

4. EXPERIMENTS

Evaluation was conducted on IARPA Babel Georgian full language pack (FLP). The FLP contains approximately 40 hours of conversational telephone speech (CTS) for training and 10 hours for development. The lexicon was obtained using the automatic approach described in [33]. The automatic speech recognition (ASR) system combines 4 diverse acoustic models in a single recognition run [34]. The diversity is obtained through the use of different model types, a tandem and a hybrid, and features, multi-lingual bottlenecks extracted by IBM and RWTH Aachen from 28 languages. The language model is a simple n-gram estimated on acoustic transcripts and web data. As a part of a larger consortium, this ASR system took part in the IARPA OpenKWS 2016 competition [35]. The development data was used to assess the accuracy of confidence estimation approaches. The data was split with a ratio of $8:1:1$ into training, validation and test sets. The ASR system was used to produce lattices. Confusion networks were obtained from lattices using consensus decoding [8]. The word error rates of the 1-best sequences are 39.9% for lattices and 38.5% for confusion networks.

The input features for the standard bi-directional recurrent neural network (BiRNN) and CN-based (BiCNRNN) are decision tree mapped posterior, duration and a 50-dimensional fastText word embedding [36] estimated from web data. The lattice-based BiRNN (BiLatRNN) makes additional use of acoustic and language model scores. All forms of BiRNNs contain one $[\overrightarrow{128}, \overleftarrow{128}]$ dimensional bi-directional LSTM layer and one 128 dimensional feed-forward hidden layer. The implementation uses PyTorch library and is available online[1]. For efficient training, model parameters are updated using Hogwild! stochastic gradient descent [37], which allows asynchronous update on multiple CPU cores in parallel.

Table 1 shows the NCE and AUC performance of confidence estimation schemes on 1-best hypotheses extracted from CNs. As expected, "raw" posterior probabilities yield poor NCE results although AUC performance is high. The decision tree, as expected, improves NCE and does not affect AUC due to the monotonicity of the mapping. The BiRNN yields gains over the simple decision tree, which is consistent with the previous work in the area [12, 13].

[1] https://github.com/qiujiali/lattice_rnn

Estimator	NCE	AUC
1-best CN posteriors	-0.1978	0.9081
+decision tree	0.2755	0.9081
+BiRNN	**0.2947**	**0.9197**

Table 1: Confidence estimation performance on 1-best CN arcs

The next experiment examines the extension of BiRNNs to confusion networks. The BiCNRNN uses a similar model topology, merges incoming arcs using the attention mechanism described in Section 3 and uses the Levenshtein algorithm with loss given by Eqn. 15 to obtain reference confidence values. The model parameters are estimated by minimising average binary cross-entropy loss on all CN arcs. The performance is evaluated over all CN arcs. When transitioning from 1-best arcs to all CN arcs the AUC performance is expected to drop due to an increase in the Bayes risk. Table 2 shows that BiCNRNN yields gains similar to BiRNN in Table 1.

Estimator	NCE	AUC
all CN posteriors	0.3105	0.8243
+decision tree	0.4659	0.8243
+BiCNRNN	**0.4970**	**0.8365**

Table 2: Confidence estimation performance on all CN arcs

As mentioned in Section 3 there are alternatives to attention for merging incoming arcs. Table 3 shows that mean and normalised posterior weights may provide a competitive alternative.[2]

Merge	NCE	AUC
max	0.4933	0.8350
mean	0.4966	0.8364
normalised posterior	0.4969	0.8363
attention	**0.4970**	**0.8365**

Table 3: Comparison of BiCNRNN arc merging mechanisms

Extending BiRNNs to lattices requires making a choice of a loss function and a method of setting reference values to lattice arcs. A simple global threshold on the amount of overlap between reference time-aligned words and lattice arcs is adopted to tag arcs. This scheme yields a false negative rate of 2.2% and false positive rate of 0.9% on 1-best CN arcs and 1.4% and 0.7% on 1-best lattice arcs. Table 4 shows the impact of using approximate loss in training the BiCNRNN. The results suggest that the mismatch between training and testing criteria, *i.e.* approximate in training and Levenshtein in testing, could play a significant role on BiLatRNN performance. Using this approximate scheme, a BiLatRNN was trained on lattices.

Table 5 compares BiLatRNN performance to "raw" posteriors and decision trees. As expected, lower AUC performances are observed due to higher Bayes risk in lattices compared to CNs. The "raw" posteriors offer poor confidence estimates as can be seen from the large negative NCE and low AUC. The decision tree yields significant gains in NCE and no change in AUC performance. Note that the AUC for a random classifier on this data is 0.2466. The BiLatRNN yields very large gains in both NCE and AUC performance.

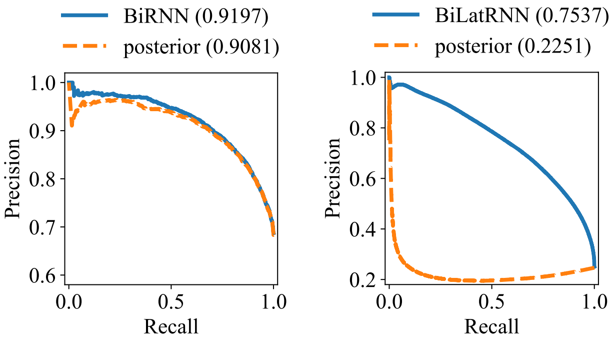
Method	NCE	AUC
Levenshtein	**0.4970**	**0.8365**
approximate	0.4873	0.8321

Table 4: Comparison of BiCNRNN arc tagging schemes

Estimator	NCE	AUC
all lattice arc posteriors	-5.0386	0.2251
+decision tree	-0.0889	0.2251
+BiLatRNN (post)	0.3880	0.7507
+BiLatRNN (attn)	**0.3921**	**0.7537**

Table 5: Confidence estimation performance on all lattice arcs

As mentioned in Section 1, applications such as language learning and information retrieval rely on confidence scores to give high-precision feedback [38] or high-recall retrieval [26, 27]. Therefore, Fig. 3 shows precision-recall curves for BiRNN in Table 1 and BiLatRNN in Table 5. Fig. 3a shows that the BiRNN yields largest gain in the region of high precision and low recall which is useful for feedback-like applications. Whereas the BiLatRNN in Fig. 3b can be seen to significantly improve precision in the high recall region, which is useful for some retrieval tasks.



(a) 1-best CN arcs

(b) all lattice arcs

Fig. 3: Precision-recall curves for Table 1 and Table 5

5. CONCLUSIONS

Confidence scores play an important role in many applications of spoken language technology. The standard form of confidence scores are decision tree mapped word posterior probabilities. A number of approaches have been proposed to improve confidence estimation, such as bi-directional recurrent neural networks (BiRNN). BiRNNs, however, can predict confidences of sequences only, which limits their more general application to 1-best hypotheses. This paper extends BiRNNs to confusion network (CN) and lattice structures. In particular, it proposes to use an attention mechanism to combine variable number of incoming arcs, shows how recursions are linked to the standard forward-backward algorithm and describes how to tag CN and lattice arcs with reference confidence values. Experiments were performed on a challenging limited resource IARPA Babel Georgian pack and shows that the extended forms of BiRNNs yield significant gains in confidence estimation accuracy over all arcs in CNs and lattices. Many related applications like information retrieval, speaker adaptation, keyword spotting and semi-supervised training will benefit from the improved confidence measure.

[2] With lattices, the attention mechanism outperforms other arc merging methods more significantly, which is reported in Table 5.

6. REFERENCES

[1] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The Microsoft 2017 conversational speech recognition system," *ICASSP*, 2018.

[2] B. Li, T. N. Sainath, A. Narayanan, J. Caroselli, M. Bacchiani, A. Misra, I. Shafran, H. Sak, G. Pundak, K. K. Chin, K. C. Sim, R. J. Weiss, K. W. Wilson, E. Variani, C. Kim, O. Siohan, M. Weintraub, E. McDermott, R. Rose, and M. Shannon, "Acoustic modeling for Google Home," in *Interspeech*, 2017.

[3] F. Wessel, R. Schluter, K Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Transactions on speech and audio processing*, 2001.

[4] H. Jiang, "Confidence measures for speech recognition: A survey," *Speech Communication*, 2005.

[5] L. F. Uebel and P. C. Woodland, "Speaker adaptation using lattice-based MLLR," in *ISCA Tutorial and Research Workshop on Adaptation Methods for Speech Recognition*, 2001.

[6] H. Y. Chan and P. C. Woodland, "Improving broadcast news transcription by lightly supervised discriminative training," in *ICASSP*, 2004.

[7] G. Tür, D. Z. Hakkani-Tür, and R. E. Schapire, "Combining active and semi-supervised learning for spoken language understanding," *Speech Communication*, 2005.

[8] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, 2000.

[9] G. Evermann and P. C. Woodland, "Large vocabulary decoding and confidence estimation using word posterior probabilities," in *ICASSP*, 2000.

[10] M. S. Seigel and P. C. Woodland, "Combining information sources for confidence estimation with CRF models," in *Interspeech*, 2011.

[11] K. Kalgaonkar, C. Liu, Y. Gong, and K. Yao, "Estimating confidence scores on ASR results using recurrent neural networks," in *ICASSP*, 2015.

[12] M. A. Del-Agua, A. Gimenez, A. Sanchis, J. Civera, and A. Juan, "Speaker-adapted confidence measures for ASR using deep bidirectional recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2018.

[13] A. Ragni, Q. Li, M. J. F Gales, and Y. Wang, "Confidence estimation and deletion prediction using bidirectional recurrent neural networks," in *SLT*, 2018.

[14] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, 2009.

[15] J. Su, Z. Tan, D. Xiong, and Y. Liu, "Lattice-based recurrent neural network encoders for neural machine translation," in *AAAI*, 2017.

[16] F. Ladhak, A. Gandhe, M. Dreyer, L. Mathias, A. Rastrow, and B. Hoffmeister, "LatticeRNN: Recurrent neural networks over lattices," in *Interspeech*, 2016.

[17] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, 1997.

[18] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, 1989.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[20] T. Schaaf and T. Kemp, "Confidence measures for spontaneous speech recognition," in *ICASSP*, 1997.

[21] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, and A. Stolcke, "Neural-network based measures of confidence for word recognition," in *ICASSP*, 1997.

[22] J. Ma and S. Matsoukas, "Unsupervised training on a large amount of Arabic broadcast news data," in *ICASSP*, 2007.

[23] M. S. Seigel and P. C. Woodland, "Detecting deletions in ASR output," in *ICASSP*, 2014.

[24] T. Mikolov, I. Sutskever, K. Chen, S. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.

[25] M. Siu, H. Gish, and F. Richardson, "Improved estimation, evaluation and applications of confidence measures for speech recognition," in *Fifth European Conference on Speech Communication and Technology*, 1997.

[26] M. J. F. Gales, K. M. Knill, and A. Ragni, "Low-resource speech recognition and keyword-spotting," in *SPECOM*, 2017.

[27] A. Ragni and M. J. F. Gales, "Automatic speech recognition system development in the "wild"," in *Interspeech*, 2018.

[28] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *ICML*, 2006.

[29] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *The 1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, 1997.

[30] G. Evermann and P. C. Woodland, "Posterior probability decoding, confidence estimation and system combination," in *The NIST 2000 Speech Transcription Workshop*, 2000.

[31] H. Xu, D. Povey, L. Mangu, and J. Zhu, "Minimum bayes risk decoding and system combination based on a recursion for edit distance," *Computer Speech & Language*, 2011.

[32] D. Povey and P. C. Woodland, "Minimum phone error and i-smoothing for improved discriminative training," in *ICASSP*, 2002.

[33] M. J. F. Gales, K. M. Knill, and A. Ragni, "Unicode-based graphemic systems for limited resource languages," in *ICASSP*, 2015.

[34] H. Wang, A. Ragni, M. J. F. Gales, K. M. Knill, P. C. Woodland, and C. Zhang, "Joint decoding of tandem and hybrid systems for improved keyword spotting on low resource languages," in *Interspeech*, 2015.

[35] A. Ragni, C. Wu, M. J. F. Gales, J. Vasilakes, and K. M. Knill, "Stimulated training for automatic speech recognition and keyword search in limited resource conditions," in *ICASSP*, 2017.

[36] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.

[37] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *NIPS*, 2011.

[38] K. M. Knill, M. J. F. Gales, K. Kyriakopoulos, A. Malinin, A. Ragni, Y. Wang, and A. Caines, "Impact of asr performance on free speaking language assessment," in *Interspeech*, 2018.