CONFIDENCE ESTIMATION FOR BLACK BOX AUTOMATIC SPEECH RECOGNITION SYSTEMS USING LATTICE RECURRENT NEURAL NETWORKS

A. Kastanos[⋆], *A. Ragni*[†,‡], *M. J. F. Gales*[⋆,‡]

[⋆]Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK
[†]Department of Computer Science, University of Sheffield, 211 Portobello, Sheffield S1 4DP, UK
ak2132@cam.ac.uk, a.ragni@sheffield.ac.uk, mjfg@eng.cam.ac.uk

ABSTRACT

Recently, there has been growth in providers of speech transcription services enabling others to leverage technology they would not normally be able to use. As a result, speech-enabled solutions have become commonplace. Their success critically relies on the quality, accuracy, and reliability of the underlying speech transcription systems. Those black box systems, however, offer limited means for quality control as only word sequences are typically available. This paper examines this limited resource scenario for confidence estimation, a measure commonly used to assess transcription reliability. In particular, it explores what other sources of word and sub-word level information available in the transcription process could be used to improve confidence scores. To encode all such information this paper extends lattice recurrent neural networks to handle sub-words. Experimental results using the IARPA OpenKWS 2016 evaluation system show that the use of additional information yields significant gains in confidence estimation accuracy.

Index Terms— confidence, sub-word, lattice, neural network

1. INTRODUCTION

Automatic Speech Recognition (ASR) has seen a surge in interest as speech enabled devices continue to proliferate the consumer market. From dedicated voice activated assistants, such as Amazon Alexa [1] and Google Home [2], to virtual assistants embedded in general purpose devices, such as Microsoft Cortana and Apple Siri, speech recognition is fast becoming a mainstream medium for interacting with technology. Though access to the underlying ASR technology has become easier [3, 4], more and more ASR systems are purchased as *black box* models in the sense that the internal state of the system is inaccessible to the user. This is particularly common in cloud-based solutions where transcriptions are often served via an application programming interface (API). The usability of these black box ASR technologies is determined by their ability to produce a correct transcription for a given audio signal. Though efforts are made to ensure they can operate over a wide variety of conditions [2], it is hard to guarantee high transcription quality for all possible scenarios. As a result, error mitigation strategies have become important.

Confidence scores provide a mechanism to mitigate error-prone ASR systems by presenting a measure of uncertainty for intelligent post-processing modules [5, 6]. These scores also find applications within upstream tasks, such as speaker adaptation [7] and semi-supervised training [8], and downstream tasks, such as machine translation [9] and information retrieval [10]. In the simplest case, confidence scores are posterior probabilities derived during the normal decoding process [11, 12]. These scores are often the only uncertainty information provided even though rich graph representations, which encode multiple hypotheses at the sub-word level, are being generated during decoding. Posterior probabilities, however, are known to over-estimate confidence [12]. Though prior work exists on improving confidence estimation [13, 12, 14, 15, 16, 17, 18], no one has examined the impact of already available information on the ability of black box ASR users to improve confidence estimates.

This paper examines confidence estimation when limited information is available. In particular, it shows that significantly more accurate estimates can be obtained if additional information is propagated by these black box systems. In order to encode complex and rich graph representations, which combine information supplied by the black box system and the user, this paper extends bi-directional lattice recurrent neural networks (BiLatRNN) [18] from the word level to include sub-word level features. Two attention-based approaches for handling variable length sub-word sequences are proposed. The more complex bi-directional encoder approach is found to be more accurate than the simpler self-attention approach [19].

The rest of this paper is organised as follows. Section 2 discusses standard representations of information within black box ASR systems, which includes graphs encoding alternative transcriptions and sub-word units. A neural network approach for encoding word level graphs is discussed in Section 3. Section 4 describes standard word level features as well as introduces two approaches for encoding sub-word information. Experimental results are presented in Section 5. The conclusions drawn from this work are given in Section 6.

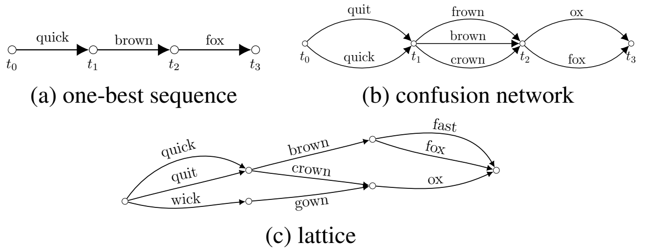
2. BLACK BOX ASR SYSTEMS

A black box ASR system is a solution provided by an external company or individual for the task of speech transcription. Such solutions are particularly popular among early-stage companies or those not primarily focused on ASR. Based on their physical location, black box ASR systems can be divided into on-premise and cloud-based. On-premise solutions physically operate on user premises to support applications with certain restrictions on security and latency of transcription. Whilst cloud-based solutions delegate transcription to a remote server that may be optimised to offer higher accuracy.

Despite many advances in speech recognition field to speed up

[‡] 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.

decoding and digital communication field to offer ultra-fast data transmission, black box ASR systems continue to provide only a very limited amount of information about the transcription process even if located on the user premises. Such information typically contains start and end times of the first and last transcribed word and the complete, one-best, word sequence. Figure 1 (a) provides a graphical representation for the one-best word sequence corresponding to the hypothetical utterance *quick brown fox*. Despite their restrictive nature, one-best sequences are the *de facto* standard output provided by commercial systems and are commonly used by downstream applications in natural language processing.



(a) one-best sequence (b) confusion network

(c) lattice

Fig. 1. Standard speech recognition output representations

Since word sequences alone carry no information about how certain the black box ASR system is, downstream applications have limited means for addressing transcription errors. The addition of simple features, such as word posterior probabilities and durations, provides the potential for a significantly better error mitigation mechanism to be devised. Other potentially useful characteristics include the word confusions found in confusion or consensus [12, 11] networks (CN) illustrated by Figure 1 (b). Such networks are typically generated by the black box ASR as a part of one-best generation process. CNs are a type of linear directed acyclical graph (DAG) which provides information pertaining to the most likely candidate transcriptions. Not only do the word confusions in CNs provide alternative word hypotheses, but they also provide an indication of the confidence in the prediction. Such rich and compact output representations have been found to be crucial for developing accurate downstream applications [20, 10], and are expected [18] to benefit confidence estimation for one-best word sequences.

CNs are normally derived from a more general DAG representation produced during decoding. These graphs, or *lattices*, illustrated by Figure 1 (c), encode a wealth of information coming from acoustic, language and pronunciation models. The clustering process behind CN construction combines multiple, not necessary precisely overlapping in time, word level lattice arcs to yield one CN arc, thus loosing the individual sources of information. Those sources can be linked to confusion network arcs and leveraged for confidence estimation if lattices were made available by black box ASR developers.

3. LATTICE RECURRENT NEURAL NETWORKS

Recently there has been interest in examining modern forms of neural networks for confidence estimation. Figure 2 shows a bi-directional recurrent neural network (BiRNN) architecture examined in [16, 17]. Given a sequence of word level feature vectors $\boldsymbol{X}_{1:T} = \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$, the BiRNN makes use of *forward* and *backward* recurrent states to predict the sequence of confidence scores $\boldsymbol{c}_{1:T} = c_1, \ldots, c_T$. The recurrent states are aimed at encoding the complete past or future information respectively. In the simplest case the forward state is defined recursively by

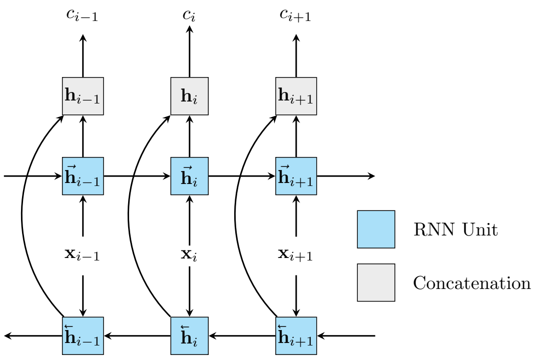

c_{i-1}
c_i
c_{i+1}
$\mathbf{h}_{i-1}$
$\mathbf{h}_i$
$\mathbf{h}_{i+1}$
$\overrightarrow{\mathbf{h}}_{i-1}$
$\overrightarrow{\mathbf{h}}_i$
$\overrightarrow{\mathbf{h}}_{i+1}$
$\mathbf{x}_{i-1}$
$\mathbf{x}_i$
$\mathbf{x}_{i+1}$
$\overleftarrow{\mathbf{h}}_{i-1}$
$\overleftarrow{\mathbf{h}}_i$
$\overleftarrow{\mathbf{h}}_{i+1}$
RNN Unit
Concatenation

Fig. 2. Bi-directional RNN for confidence prediction

$$\overrightarrow{\boldsymbol{h}}_i = \boldsymbol{\sigma}(\boldsymbol{W}^{(\overrightarrow{h})}\overrightarrow{\boldsymbol{h}}_{i-1} + \boldsymbol{W}^{(x)}\boldsymbol{x}_i) \tag{1}$$

where $\boldsymbol{W}^{(\overrightarrow{h})}$ and $\boldsymbol{W}^{(x)}$ are weight matrices for the forward state and the feature vector respectively, $\boldsymbol{\sigma}(\cdot)$ is an element-wise non-linearity, such as sigmoid, $\overrightarrow{\boldsymbol{h}}_0$ can be set to $\mathbf{0}$ or learnt. The backward state can be defined analogously. Given the forward and backward states at time i, the confidence score can be predicted by

$$c_i = \sigma(\boldsymbol{w}^{(c)^{\mathsf{T}}}\boldsymbol{h}_i + b^{(c)}) \tag{2}$$

where $\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}}_i^{\mathsf{T}} \ \overleftarrow{\boldsymbol{h}}_i^{\mathsf{T}}]^{\mathsf{T}}$, $\boldsymbol{w}^{(c)}$ and $b^{(c)}$ are weight vectors and a scalar bias, σ is a non-linearity mapping confidence scores to $[0, 1]$ range. Note that unlike many other supervised learning problems, the targets for confidence scores need to be derived by automatically aligning predicted and manually transcribed word sequences [18].

The BiRNN is inherently limited to sequence data. As discussed in Section 2 one-best sequences carry only a small portion of information otherwise available in either constrained (CN) or unconstrained (lattice) DAG format. Those DAGs are highly flexible structures that can be additionally enriched with a wide range of features [21, 22]. Recently there has been much interest in examining neural network extensions to DAGs and other general graph structures [23, 18, 24]. The key question that any such approach needs to answer is how information associated with multiple graph arcs or nodes is combined. Figure 3 illustrates one such bi-directional approach for lattices (BiLatRNN). Compared to the sequence model in Figure 2, the lattice model has one or more past recurrent states which propagates the current state to one or more subsequent states. In order to handle a variable number of past recurrent states, BiLatRNN makes use of an attention mechanism to create a combined representation

$$\overrightarrow{\boldsymbol{h}}_{\overrightarrow{\mathcal{N}}_i} = \sum_{j \in \overrightarrow{\mathcal{N}}_i} \alpha_j \overrightarrow{\boldsymbol{h}}_j \tag{3}$$

where $\overrightarrow{\mathcal{N}}_i$ is a set of incoming arcs for arc i. The attention mechanism makes use of arc contributions $\boldsymbol{e}$ to yield attention weights

$$\alpha_j = \exp(e_j) \Big/ \sum_{j' \in \overrightarrow{\mathcal{N}}_i} \exp(e_{j'}) \tag{4}$$

There are numerous ways for how arc contributions can be defined, such as scaled dot-product self-attention [19]

$$e_j = \overrightarrow{\boldsymbol{h}}_j^{\mathsf{T}} \left[\frac{1}{\sqrt{\dim(\overrightarrow{\boldsymbol{h}}_j)}} \boldsymbol{I} \right] \overrightarrow{\boldsymbol{h}}_j \tag{5}$$

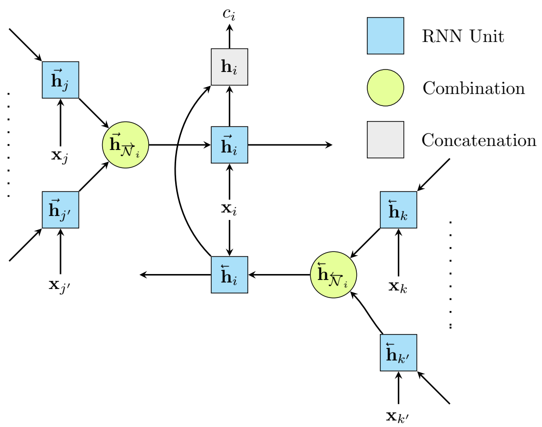


Fig. 3. Bi-directional lattice RNN for confidence prediction

multiplicative self-attention [25] with trainable weights $\boldsymbol{W}^{(m)}$,

$$e_j = \overrightarrow{\boldsymbol{h}}_j^{\mathsf{T}} \boldsymbol{W}^{(m)} \overrightarrow{\boldsymbol{h}}_j \tag{6}$$

or additive attention [25, 26] with trainable weights $\boldsymbol{W}^{(q)}$ and $\boldsymbol{w}^{(a)}$.

$$e_j = \sigma\left(\boldsymbol{w}^{(a)\mathsf{T}} \boldsymbol{\sigma}\left(\boldsymbol{W}^{(q)} \begin{bmatrix} \boldsymbol{k}_j^{\mathsf{T}} & \overrightarrow{\boldsymbol{h}}_j^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}\right)\right) \tag{7}$$

The additive form offers flexibility by "querying" the state $\overrightarrow{\boldsymbol{h}}_j$ with a key $\boldsymbol{k}_j$. In previous work [18] the key was set to

$$\boldsymbol{k}_j = \begin{bmatrix} \log(\hat{c}_j) & \log(\hat{\mu}_j) & \log(\hat{\sigma}_j) \end{bmatrix}^{\mathsf{T}} \tag{8}$$

where $\hat{c_j}$, $\hat{\mu}_j$ and $\hat{\sigma}_j$ are posterior probability, mean and standard deviation of all arc posterior probabilities which overlap in time with arc j. Such key design should enable the attention mechanism to downweight states of unlikely paths. Once the combined representation have been obtained, the current state can be updated by

$$\overrightarrow{\boldsymbol{h}}_i = \boldsymbol{\sigma}(\boldsymbol{W}^{(\overrightarrow{h})} \overrightarrow{\boldsymbol{h}}_{\overrightarrow{\mathcal{N}}_i} + \boldsymbol{W}^{(x)} \boldsymbol{x}_i) \tag{9}$$

The confidence score prediction is then done using equation (2). The targets for lattice arc confidence scores are generated by extending the alignment algorithm for one-best sequences as described in [18].

4. FEATURES

As discussed in Section 2, a large amount of information is produced during the decoding process. However, for users of black box ASR typically only the one-best word sequence $\boldsymbol{w}_{1:T} = w_1, \ldots, w_T$ is available. If posterior probabilities and durations were also propagated the complete set of word level features could be expressed as

$$\boldsymbol{x}_i^{(w)} = \begin{bmatrix} \boldsymbol{e}_{w_i}^{\mathsf{T}} & d_{w_i} & \log(\hat{c}_{w_i}) \end{bmatrix}^{\mathsf{T}} \tag{10}$$

where $\boldsymbol{e}_{w_i}$ is word w_i represented as a one-hot encoding or embedding, d_{w_i} is the word duration, and $\hat{c}_{w_i}$ is the posterior probability. The word embedding is a continuous word representation [27] that can either be trained jointly with the rest of the neural network or independently on large quantities of text data [28, 29] and then possibly fine-tuned. These simple features have been used with both BiRNN [17] and BiLatRNN [18] for confidence prediction.

As mentioned in Section 3, a wide range of additional information can be augmented to graph structures such as confusion networks and lattices. Any word level information can be added by simply extending the number of features in equation (10). The use of sub-word information, such as phone, grapheme, morpheme, or byte-pair encoding, is more complicated due to variable length nature of sub-word sequences. A fixed length representation can be obtained by adopting the attention mechanism described in Section 3

$$\boldsymbol{x}_i^{(s)} = \sum_{j \in \mathcal{S}_i} \alpha_{i,j} \boldsymbol{h}_{i,j} \tag{11}$$

where $\mathcal{S}_i$ is a sequence of sub-word units for word w_i, $\alpha_{i,j}$ and $\boldsymbol{h}_{i,j}$ are an attention weight and continuous representation for sub-word unit s_j respectively. There are several options for how sub-word representations $\boldsymbol{h}_{i,j}$ can be derived. In the simplest case, sub-word features can be defined in a similar manner to the word features by

$$\boldsymbol{x}_{i,j} = \begin{bmatrix} \boldsymbol{\epsilon}_{s_j}^{\mathsf{T}} & d_{s_j} & \log(\hat{c}_{s_j}) \end{bmatrix}^{\mathsf{T}} \tag{12}$$

where $\boldsymbol{\epsilon}_{s_j}$ is either a one-hot encoding or embedding, d_{s_j} and $\hat{c}_{s_j}$ are duration and posterior probability respectively for sub-word s_j. A more powerful approach would be to use a bi-directional *encoder* as shown in Figure 4. To estimate sub-word attention weights $\boldsymbol{\alpha}_i$ for each word w_i it is possible to use one of the approaches discussed in Section 3. For instance, the use of additive attention in equation (7) offers a number of interesting choices for selecting keys $\boldsymbol{k}_{i,j}$ to match against hidden states $\boldsymbol{h}_{i,j}$ that may include not only sub-word but also word level information.

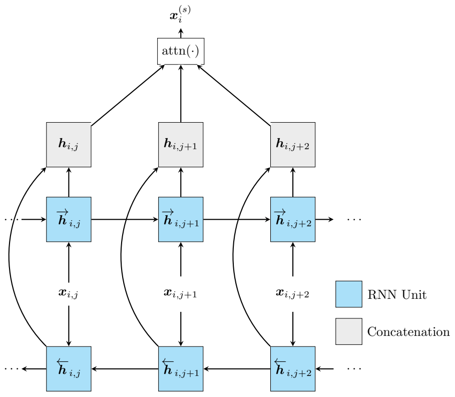

$\boldsymbol{x}_i^{(s)}$
attn(·)
$\boldsymbol{h}_{i,j}$
$\boldsymbol{h}_{i,j+1}$
$\boldsymbol{h}_{i,j+2}$
$\overrightarrow{\boldsymbol{h}}_{i,j}$
$\overrightarrow{\boldsymbol{h}}_{i,j+1}$
$\overrightarrow{\boldsymbol{h}}_{i,j+2}$
$\boldsymbol{x}_{i,j}$
$\boldsymbol{x}_{i,j+1}$
$\boldsymbol{x}_{i,j+2}$
$\overleftarrow{\boldsymbol{h}}_{i,j}$
$\overleftarrow{\boldsymbol{h}}_{i,j+1}$
$\overleftarrow{\boldsymbol{h}}_{i,j+2}$
RNN Unit
Concatenation

Fig. 4. Bi-directional RNN sub-word encoder

As discussed in Section 2, one-best word sequences are usually obtained from CNs rather than lattices, which makes information encoded into the latter not directly available for the former. Therefore, all prior work examined confidence estimation either based on CN or lattice output. Lattices, however, provide a rich and flexible framework for representing not only already available information but also various other external sources. As a result, they naturally facilitate interesting and often powerful features. Two of the simplest lattice features are the acoustic model score and the language model score [30]. More intricate features include acoustic stability [31] and hypothesis density [32]. In order to make those features available, an alignment of CNs to lattices can be performed to match each CN arc to one or more lattice arcs depending on the time overlap tolerance specified. Provided that the lattices are large enough, the chance that any given CN arc cannot be matched to at least one lattice arc is small. Once lattice arcs have been aligned with CN arcs a number of approaches, such as simple averaging or an attention mechanism, can be used to yield lattice features for CN arcs.

5. EXPERIMENTS

The experiments in this work were conducted on the decoded output from a graphemic ASR system trained for the IARPA OpenKWS 2016 competition. The audio recordings consist of Georgian conversational telephone speech, of which 25 hours was used for training and testing confidence estimation approaches. The predictions from this system, which are treated as a black box, are split into independent training, cross-validation, and test sets with an $8:1:1$ ratio. After CN decoding, the ASR system has a word error rate of approximately 38%, which leads to an imbalanced distribution of correct and incorrect word predictions. All models, BiRNN and BiLatRNN, use a single 128-dimensional bi-directional LSTM layer with a fully connected hidden layer consisting of 128 neurons. The subword encoder uses a similar architecture based on a 10-dimensional GRU layer. The results are presented in terms of two standard metrics, normalised cross-entropy (NCE), which indicates the relative change in cross-entropy when the empirical estimate of correctness is replaced with hypothesised confidence score [17, 33], and area under the curve (AUC), where the precision-recall curve is used to mitigate the effect of the dataset imbalance [34]. A random classifier in this setup will render an AUC score of 0.6310.

The set of word level features used include a 50-dimensional `fastText` [35] pre-trained word embedding, duration, and posterior without and with decision tree mapping [12]. Table 1 demonstrates how the incremental addition of these input features to the BiRNN model results in performance improvements relative to the use of simple word embeddings. As expected, the use of words and durations yields low, although higher than random, AUC values whilst the introduction of posteriors sees a large performance improvement.

Word Features	NCE	AUC
words	0.0358	0.7496
+duration	0.0541	0.7670
+ posteriors	0.2765	0.9033
+ mapping	**0.2911**	**0.9121**

Table 1. Confidence estimation performance using word features

The set of sub-word level features used included a 4-dimensional `word2vec` [27] pre-trained grapheme embedding and duration. As described in Section 4, sub-word features can be incorporated into word level models using an attention mechanism applied either directly to sub-word features or to encoder states. A comparison of attention approaches (see Section 3), which is not reported here due to space constraints, showed that the additive attention with the subword embedding and duration as a key yielded slightly better results and hence is used in the rest of this section. Table 2 shows that the use of sub-word information (embedding, +duration) and more complex representations (+encoder) yields small but consistent gains.

Sub-word Features	NCE	AUC
none	0.2911	0.9121
embedding	0.2936	0.9127
+ duration	0.2944	0.9129
+encoder	**0.2978**	**0.9139**

Table 2. Impact of sub-word features

The BiRNN examined so far lacked any information about competing transcriptions available within CNs. Depending on application there are several ways how such information can be utilised. If the task is to predict confidence scores for one-best word sequences (as in this work) the training loss should be accumulated over one-best sequences only. However, if confidence scores of all arcs are of interest (as in previous work [18]) the training loss should be accumulated over all arcs. Note that the forward propagation is done through all arcs irrespective of the choice made above. Table 3 shows that although both BiLatRNN approaches yield gains over the one-best baseline, the former as expected yields better AUC values. Table 4 also shows that word confusion and sub-word information are quite complimentary, yielding significant gains over word only one-best baseline.

Confusions	Loss	NCE	AUC
1-best	1-best	0.2911	0.9121
CN	1-best	0.2931	0.9201
CN	CN	0.2934	0.9178

Table 3. Impact of word confusion information

Features	NCE	AUC
word (all)	0.2911	0.9121
+confusions	0.2934	0.9201
+sub-word	0.2998	0.9228
+lattice	**0.3004**	**0.9231**

Table 4. Impact of word confusion, sub-word and lattice features

As discussed in Section 4, a range of lattice features can be incorporated into BiLatRNNs by aligning lattices to CNs. As a proof of concept this work examined a simple set of lattice features: the acoustic and language model scores. Given a relatively large set of lattices and a tight threshold on time overlap, the alignment process failed to match 1.7% of training utterances, which, in this work, were removed from training. Note that significantly larger lattices can be obtained by simply increasing decoding beam size. Table 4 shows that the BiLatRNN can leverage even the simplest of lattice features with more gains expected from more complex approaches.

6. CONCLUSION

With black box automatic speech recognition (ASR) systems becoming more popular, the importance of error mitigation strategies grows. Despite clear evidence from the literature that word sequences alone are not adequate for building accurate applications, the restricted form of one-best remains the *de facto* standard output of commercial ASR. Word sequences, however, provide a limited opportunity for devising even the simplest error mitigation strategy, a confidence score. This paper examines a hypothetical scenario where progressively more (normally discarded) information, such as confusion networks and lattices, are propagated to the user. To leverage these graph structures a bi-directional lattice recurrent neural network was used and extended to handle sub-word information. Experimental results on the challenging IARPA OpenKWS 2016 task show that additional information is crucial and can be easily leveraged using available neural network approaches.

7. REFERENCES

[1] A. Ram, R. Prasad, C. Khatri, A. Venkatesh, R. Gabriel, Q. Liu, J. Nunn, B. Hedayatnia, M. Cheng, A. Nagar, E. King, K. Bland, A. Wartick, Y. Pan, H. Song, S. Jayadevan, G. Hwang, and A. Pettigrue, "Conversational AI: The science behind the Alexa prize," in *arXiv preprint* arXiv:1801.03604, 2018.

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

[3] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, A. Ragni, V. Valtchev, P. C. Woodland, and C. Zhang, *The HTK Book*, University of Cambridge, http://htk.eng.cam.ac.uk, 2015.

[4] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *ASRU*, 2011.

[5] Timothy J Hazen, Stephanie Seneff, and Joseph Polifroni, "Recognition confidence scoring and its use in speech understanding systems," *Computer Speech & Language*, vol. 16, no. 1, pp. 49–67, 2002.

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

[7] L. F. Uebel and P. C. Woodland, "Speaker adaptation using lattice-based MLLR," in *ITRW*, 2001.

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

[9] S. Saleem, S.-C. Jou, S. Vogel, and T. Schultz, "Using word lattice information for a tighter coupling in speech translation systems," in *ICSLP*, 2004.

[10] R. Zbib, L. Zhao, D. Karakos, W. Hartmann, J. DeYoung, Z. Huang, Z. Jiang, N. Rivkin, L. Zhang, R. Schwartz, and J. Makhoul, "Neural-network lexical translation for cross-lingual IR from text and speech," in *SIGIR*, 2019, pp. 645–654.

[11] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus among words: Lattice-based word error minimization," in *Eurospeech*, 1999.

[12] G. Evermann and P. C. Woodland, "Large vocabulary decoding and confidence estimation using word posterior probabilities," in *ICASSP*, 2000, vol. 3, pp. 1655–1658.

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

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

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

[16] 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 TASLP*, vol. 26, no. 7, pp. 1198–1206, 2018.

[17] A. Ragni, Q. Li, M. J. F. Gales, and Y. Wang, "Confidence estimation and deletion prediction using bidirectional recurrent neural networks," in *SLT*, 2018, pp. 204–211.

[18] Q. Li, P. M. Ness, A. Ragni, and M. J. F. Gales, "Bi-directional lattice recurrent neural networks for confidence estimation," in *ICASSP*, 2019.

[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, pp. 6000–6010.

[20] V.-B. Le, L. Lamel, A. Messaoudi, W. Hartmann, J.-L. Gauvain, C. Woehrling, J. Despres, and A. Roy, "Developing STT and KWS systems using limited language resources," in *Interspeech*, 2014.

[21] G. Zweig and P. Nguyen, "A segmental CRF approach to large vocabulary continuous speech recognition," in *ASRU*, 2009.

[22] A. Ragni and M. J. F. Gales, "Derivative kernels for noise robust ASR," in *ASRU*, 2011, pp. 119–124.

[23] F. Ladhak, A. Gandhe, M. Dreyer, L. Mathias, A. Rastrow, and B. Hoffmeister, "LatticeRNN: Recurrent neural networks over lattices," in *Interspeech*, 2016, pp. 695–699.

[24] P. Zhang, B. Chen, N. Ge, and K. Fan, "Lattice transformers for speech translation," in *ACL*, 2019, pp. 6475–6484.

[25] T. Luong, H. Pham, and C. Manning, "Effective approaches to attention-based neural machine translation," in *EMNLP*, 2015.

[26] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[27] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *arXiv preprint arXiv:1310.4546*, 2013.

[28] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *NAACL*, 2018.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[30] J. Pinto and R. N. V. Sitaram, "Confidence measures in speech recognition based on probability distribution of likelihoods," in *Interspeech*, 2005.

[31] 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.

[32] T. Kemp and T. Schaaf, "Estimating confidence using word lattices," in *Eurospeech*, 1997, pp. 827–830.

[33] M.-H. Siu, H. Gish, and F. Richardson, "Improved estimation, evaluation and applications of confidence measures for speech recognition," in *Eurospeech*, 1997.

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

[35] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," in *ACL*, 2017.