Confidence Estimation for Black Box Automatic Speech Recognition Systems using Lattice Recurrent Neural Networks

ICASSP 2020

A. Kastanos*, A. Ragni*†, M.J.F. Gales*

April 15, 2020

* Dept of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK
† Dept of Computer Science, University of Sheffield, 211 Portobello, Sheffield S1 4DP, UK

Introduction

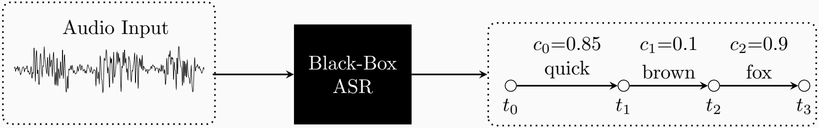

Figure 1: Overview of a black-box ASR system

- Cloud-based ASR solutions are becoming the norm
  - Increasing complexity of ASR
  - Fewer companies can afford to build their own systems
  - The internal states of *black-box* systems are inaccessible
- Word-based confidence scores are an indication of reliability

Speech Recognition and Confidence Scores

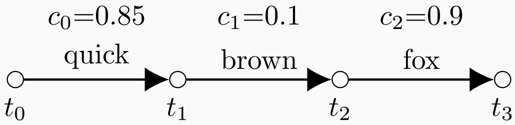

Figure 2: One-best word sequence with a word-level confidence score

How do we typically obtain confidence scores?

- Word posterior probability - known to be overly confident [1]
- Decision tree mapping requires calibration
- Can we do better?

Deep Learning for Confidence Estimation

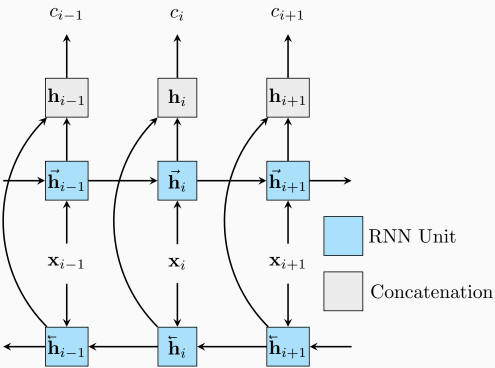

Figure 3: Bi-directional RNN for confidence prediction on one-best sequences

- Bi-directional RNN to predict if each word is correct
    - What kind of features are available?
    - What if we have access to complicated structures?

Features

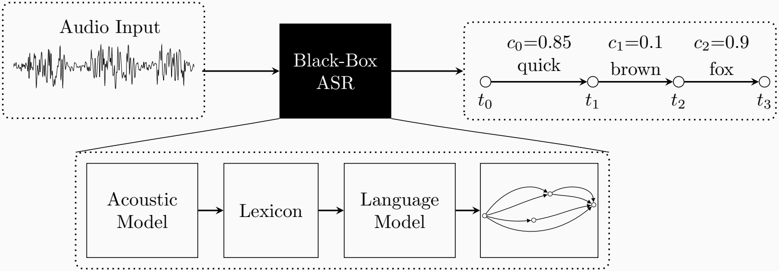

Figure 4: Detailed look at ASR features

Can we extract these features?

- Sub-word level information
- Competing hypotheses
- Lattice features

Sub-word Unit Encoder

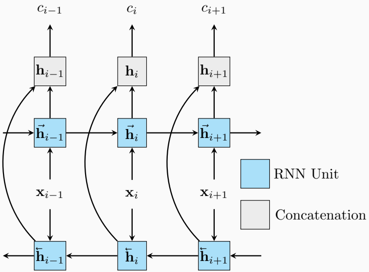

Figure 5: Word confidence classifier

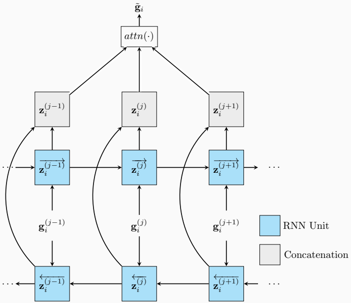
$\tilde{\mathbf{g}}_i$
$attn(\cdot)$
$\mathbf{z}_i^{(j-1)}$
$\mathbf{z}_i^{(j)}$
$\mathbf{z}_i^{(j+1)}$
$\overrightarrow{\mathbf{z}_i^{(j-1)}}$
$\overrightarrow{\mathbf{z}_i^{(j)}}$
$\overrightarrow{\mathbf{z}_i^{(j+1)}}$
$\mathbf{g}_i^{(j-1)}$
$\mathbf{g}_i^{(j)}$
$\mathbf{g}_i^{(j+1)}$
$\overleftarrow{\mathbf{z}_i^{(j-1)}}$
$\overleftarrow{\mathbf{z}_i^{(j)}}$
$\overleftarrow{\mathbf{z}_i^{(j+1)}}$
$\cdots$
RNN Unit
Concatenation

Figure 6: Sub-word feature extractor

- Given a lexicon, we can extract grapheme features
- `fox` → { `f`, `o`, `x` }
- Convert a variable length grapheme sequence into a fixed size
- Deep learning to aggregate features

Alternative Hypothesis Representations

An intermediate step in generating a one-best sequence is the generation of **lattices**.

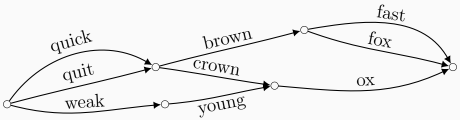

Figure 7: Lattice

From lattices, we can obtain **confusion networks** by clustering arcs.

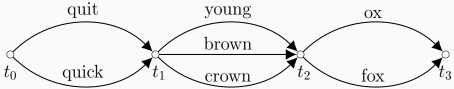

Figure 8: Confusion network

How do we handle non-sequential models?

Lattice Recurrent Neural Networks

A generalisation of bi-directional RNNs to handle multiple incoming arcs:

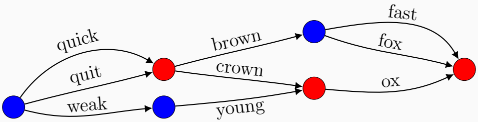

Figure 9: Red nodes have multiple incoming arcs, while blue nodes only have one.

Attention to learn relative importance [2]:

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

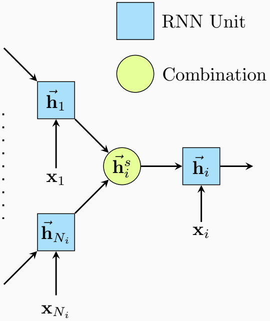

Figure 10: Arc merging mechanism as implemented by LatticeRNN [3]

Extracting Lattice Features

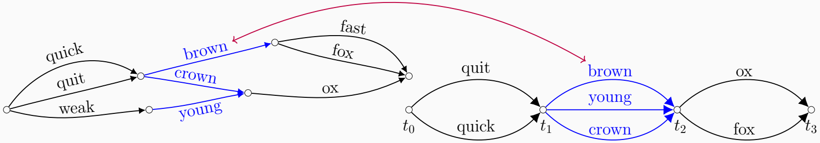

Figure 11: Arc matching

- Match arcs to the corresponding lattice arc
- What kind of features could we extract?
  - Acoustic and Language model scores
  - Lattice embeddings
  - Hypothesis density

Experiments (One-best)

Large gains are obtained by introducing additional information.

Features		NCE	AUC
word	words	0.0358	0.7496
	+duration	0.0541	0.7670
	+ posteriors	0.2765	0.9033
	+ mapping	0.2911	0.9121
sub-word	+ embedding	0.2936	0.9127
	+ duration	0.2944	0.9129
	+encoder	**0.2978**	**0.9139**

Table 1: Impact of word and sub-word features. IARPA BABEL Georgian (25 hours).

Experiments (Confusion Networks)

Significant gains from alternative hypotheses and basic lattice features.

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 2: Impact of competing hypothesis information. IARPA BABEL Georgian (25 hours).

Conclusion

- Prevalence of black-box ASR
    - Limited ability to assess transcription reliability
- Confidence estimates can be improved by providing available information
    - Deep learning approach for incorporating sub-word features
    - Deep learning framework for introducing lattice features

References

G. Evermann and P.C. Woodland,
“Posterior probability decoding, confidence estimation and system combination,” 2000.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin,
“Attention is all you need,”
in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

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

Thank you

Figure 12: Source code: `https://github.com/alecokas/BiLatticeRNN-Confidence`