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
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
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

Can we extract these features?

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

Figure 4: Detailed look at ASR features
Given a lexicon, we can extract grapheme features

- \( \text{fox} \rightarrow \{ f, o, x \} \)
- Convert a variable length grapheme sequence into a fixed size
- Deep learning to aggregate features
An intermediate step in generating a one-best sequence is the generation of lattices.

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

How do we handle non-sequential models?
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]:

\[ \vec{h}_i = \sum_{j \in \hat{N}_i} \alpha_j \vec{h}_j \]
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
Large gains are obtained by introducing additional information.

<table>
<thead>
<tr>
<th>Features</th>
<th>NCE</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>word</td>
<td>0.0358</td>
<td>0.7496</td>
</tr>
<tr>
<td>+ duration</td>
<td>0.0541</td>
<td>0.7670</td>
</tr>
<tr>
<td>+ posteriors</td>
<td>0.2765</td>
<td>0.9033</td>
</tr>
<tr>
<td>+ mapping</td>
<td>0.2911</td>
<td>0.9121</td>
</tr>
<tr>
<td>sub-word + embedding</td>
<td>0.2936</td>
<td>0.9127</td>
</tr>
<tr>
<td>+ duration</td>
<td>0.2944</td>
<td>0.9129</td>
</tr>
<tr>
<td>+ encoder</td>
<td>0.2978</td>
<td>0.9139</td>
</tr>
</tbody>
</table>

*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.

<table>
<thead>
<tr>
<th>Features</th>
<th>NCE</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>word (all)</td>
<td>0.2911</td>
<td>0.9121</td>
</tr>
<tr>
<td>+confusions</td>
<td>0.2934</td>
<td>0.9201</td>
</tr>
<tr>
<td>+sub-word</td>
<td>0.2998</td>
<td>0.9228</td>
</tr>
<tr>
<td>+lattice</td>
<td>0.3004</td>
<td>0.9231</td>
</tr>
</tbody>
</table>

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


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