Ensemble combination between different time segmentations

Jeremy Wong, Dimitrios Dimitriadis, Kenichi Kumatani, Yashesh Gaur, George Polovets, Partha Parthasarathy, Eric Sun, Jinyu Li, and Yifan Gong

Microsoft Speech and Language Group
Ensemble combination

- Hypothesis-level combination assumes that all models use the same input time segments.

Propose:
- Method to allow different input segmentation times between models.
Applications for different time segmentations

• Combination between different time segmentations can be used for:
  ➢ Different VAD front-ends for each model.
  ➢ Audio from multiple unsynchronised recording devices.
  ➢ Overlapping inference.
  ➢ Using a 1\textsuperscript{st} pass ASR to refine the time segmentations for a 2\textsuperscript{nd} pass ASR.
Meeting transcription setup

• 1\textsuperscript{st} pass streaming ASR -> diarisation -> 2\textsuperscript{nd} pass offline ASR
• 1\textsuperscript{st} pass ASR uses VAD segments.
• 2\textsuperscript{nd} pass ASR uses per-speaker segments.
• Want to combine 1\textsuperscript{st} pass and 2\textsuperscript{nd} pass ASR hypotheses to improve 2\textsuperscript{nd} pass performance.
1. Convert N-best list into confusion network.
2. Estimate start and end times of each confusion set.
3. Estimate the speaker ID for each confusion set from the 1-best hypothesis.
4. Split up confusion network into separate confusion sets.
5. Re-join consecutive confusion sets to match time segments.
6. Do Confusion Network Combination (CNC) between all models.

Confusion network splitting

**Advantages:**
- 1-best is preserved after splitting and re-joining.

**Disadvantages:**
- Start and end times of each confusion set are approximate.
- Word sequence context of language model scores is not preserved.
1. Distribute hypothesis scores to words.
2. Estimate the speaker ID for each N-best word from the 1-best hypothesis.
3. Split up the N-best lists.
4. Re-join N-best lists according to segment times.
5. Do Minimum Bayes' Risk (MBR) combination between all models.
N-best list splitting

**Advantages:**
- Exact word start and end times are preserved from ASR decoding.
- Word sequence context of language model scores is preserved.

**Disadvantages:**
- 1-best may not be preserved after splitting and re-joining.
Distribute hypothesis scores to words

- Black-box ASR system may only produce per-hypothesis scores.
- Estimate per-word scores by:
  1. Convert N-best list to prefix and suffix trees.
  2. Push weights to branches.
  3. Take log-average of per-word scores from prefix and suffix trees.
- Prefix and suffix trees concentrate weights at opposite ends.
Experiments

Dataset:

• Internal Microsoft meetings.
• \textit{dev} set: 51 meetings, 23 hours
• \textit{eval} set: 60 meetings, 35 hours
• \textit{Average of 7 participants per meeting.}

Speaker-attributed WER Metric:

• For each speaker, compute the WER of that speaker’s hypothesis vs reference.
• Average the WERs over all speakers.
Experiments

Models:

• **1st pass hybrid**: streaming latency-controlled and layer-trajectory BLSTM.
• **2nd pass hybrid**: ensemble of 2 offline BLSTMs.
• **2nd pass LAS**: offline BLSTM encoder, LSTM decoder.
• **Hybrid LM**: 5-gram + NNLM

**N-best list size: 16**
Score distribution method

- Distribute hypothesis-level scores to words for streaming 1\textsuperscript{st} pass model.
- Split and re-join 1\textsuperscript{st} pass N-best lists to match 2\textsuperscript{nd} pass segments.

<table>
<thead>
<tr>
<th>Split</th>
<th>Per-word scores</th>
<th>eval Speaker-attributed WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>original</td>
<td>20.43</td>
</tr>
<tr>
<td></td>
<td>original</td>
<td>22.09</td>
</tr>
<tr>
<td></td>
<td>language model re-score</td>
<td>22.09</td>
</tr>
<tr>
<td>yes</td>
<td>prefix tree</td>
<td>20.62</td>
</tr>
<tr>
<td></td>
<td>suffix tree</td>
<td>20.60</td>
</tr>
<tr>
<td></td>
<td>log-average</td>
<td>20.55</td>
</tr>
</tbody>
</table>

- After splitting, log-average between prefix and suffix trees performs best.
- Splitting yields degradation.
Multi-pass combination

• Single model performance.

<table>
<thead>
<tr>
<th>Segments</th>
<th>Model</th>
<th>Speaker-attributed WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>dev</td>
</tr>
<tr>
<td>1&lt;sup&gt;st&lt;/sup&gt; pass</td>
<td>streaming hybrid</td>
<td>21.43</td>
</tr>
<tr>
<td>2&lt;sup&gt;nd&lt;/sup&gt; pass</td>
<td>streaming hybrid</td>
<td>20.87</td>
</tr>
<tr>
<td></td>
<td>offline hybrid</td>
<td>19.93</td>
</tr>
<tr>
<td></td>
<td>offline LAS</td>
<td>19.91</td>
</tr>
</tbody>
</table>

• Offline model outperforms streaming model.
• 2<sup>nd</sup> pass segments yield gains over 1<sup>st</sup> pass segments for the same model.
Multi-pass combination

<table>
<thead>
<tr>
<th>Segments</th>
<th>Model</th>
<th>Speaker-attributed WER (%)</th>
<th>Combination between 1st and 2nd pass hypotheses</th>
<th>Speaker-attributed WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st pass</td>
<td>streaming hybrid</td>
<td>21.43 20.43</td>
<td>CNC streaming hybrid + offline hybrid</td>
<td>20.01 19.10</td>
</tr>
<tr>
<td>2nd pass</td>
<td>streaming hybrid</td>
<td>20.87 19.96</td>
<td>CNC streaming hybrid + offline LAS</td>
<td>19.71 18.71</td>
</tr>
<tr>
<td></td>
<td>offline hybrid</td>
<td>19.93 19.13</td>
<td>MBR streaming hybrid + offline hybrid</td>
<td>19.83 19.00</td>
</tr>
<tr>
<td></td>
<td>offline LAS</td>
<td>19.91 19.04</td>
<td>MBR streaming hybrid + offline LAS</td>
<td>19.30 18.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MBR offline hybrid + offline LAS</td>
<td>19.11 18.24</td>
</tr>
</tbody>
</table>

- MBR with N-best splitting outperforms CNC with confusion network splitting.
- Offline hybrid + offline LAS performs best, but is computationally expensive.
- Streaming hybrid + offline LAS yields reasonable gains, with only single model in 2nd pass.
- Hybrid + LAS outperforms hybrid + hybrid, suggesting greater diversity.
- Streaming hybrid (on 2nd pass segments) + offline hybrid eval WER = 18.37 %.
Summary

• **Proposed:**
  - Allow different time segments in combination by splitting and re-joining of N-best lists.
  - Estimate per-word scores from per-hypothesis scores using trees.

• Improve 2nd pass performance without additional computational cost.

• Showed that hybrid + LAS outperforms hybrid + hybrid.