A Keyword-Aware Grammar Framework for LVCSR-based Spoken Keyword Search

I-Fan Chen, Chongjia Ni, Boon Pang Lim, Nancy F. Chen, and Chin-Hui Lee
Introduction

- Spoken keyword search (KWS)
  - To detect a set of preselected keywords in continuous speech

- LVCSR-based KWS
  - Search keywords in LVCSR transcribed text documents
  - Use \( n \)-gram LM grammar
    - Comparing to the simple keyword-filler loop grammar
  - Pros
    - Less false alarms, Better performance, Flexible on keywords
  - Cons
    - \( n \)-gram LMs need a great amount of training data to cover potential keywords
    - High miss rates for uncovered keywords
      - Whose prior probabilities are seriously underestimated
Keyword-aware grammar framework

- Integrate keyword-filler loop grammars into LVCSR-based KWS [1]
  - Boost KW probabilities in the decoding grammar
- Approximate the boosting effect
  - Context-Simulated keyword language model (CS-KWLM) interpolated LMs
    - Train a context-simulated KWLM using the system keyword list
    - Interpolate the CS-KWLM with the original n-gram LM
    - \[ P_{\text{INT}_\text{LM}}(w \mid h) = \alpha \cdot P_{\text{CS-KWLM}}(w \mid h) + (1 - \alpha)P_{\text{LM}}(w \mid h) \]
    - \( \alpha \in [0, 1] \) controls the weight of CS-KWLM in the interpolation

Exact Realization of the KW-aware Grammar

- **Weighted Finite State Automata (WFSA)**
  - Insert KW paths to the \( n \)-gram LM grammar WFSA of the system.
  - For each \( kw = w_1 \ldots w_L \)
    - \( P_{KW\text{-aware}}(kw \mid h) = \max \left\{ P_{n\text{-gram}}(kw \mid h), \kappa \right\} \)
    - \( P_{n\text{-gram}}(kw \mid h) = \prod_{i=1}^{L} P(w_i \mid h_i) \)
    - \( \kappa \): prior constant
      - global constant (or \( \kappa_c \) for KW in class \( c \))

- **Disambiguation**
  - Non-deterministic
    - Both \( n \)-gram grammar and the inserted standalone KW paths can represent the keywords
  - Use disambiguation symbol \( \#k \) (or \( \#k_c \) for KW in class \( c \))
    - For arcs entering the standalone KW path
Realize the KW-aware Grammar using WFSA

- Insert a keyword path for a 3-word keyword \( w_1w_2w_3 \)
- In this work
  - A global \( \kappa \) and a single keyword initial state were used.
Experimental Setup (1)

- IARPA Babel OpenKWS13 Vietnamese and OpenKWS14 Tamil Limited Language Pack tasks
  - Conversational speech over telephone channels (landline, cell-phone, etc.) with sampling rate 8 kHz
- Training data
  - 10-hour transcribed audio data
- Development data
  - 2-hour subset of 10-hour IARPA development data
- Evaluation data
  - 15-hour evaluation part 1 (evalpart1) data
Experimental Setup (2)

- **Evaluation keyword list (official)**
  - Vietnamese: 4065 keywords
    - Keyword lengths range from 1 to 6 words
  - Tamil: 5576 keywords
    - Keyword lengths range from 1 to 5 words

- **Performance measure**
  - Number of miss detections
  - Actual Term Weight Value (ATWV)
    - A metric takes both miss and false alarm rate into account

- **Baseline LVCSR-based KWS system**
  - Kaldi Babel recipe
    - **Acoustic Feature** – BNF + fMLLR [ on top of PLP+F0 ]
    - **Acoustic models** – DNN model with sMBR training
    - **Language models** – 3-gram LM trained with the 10-hour transcription
Comparison of grammar WFSAs

- Vietnamese task [4065 keywords]
- KW-aware grammar WFSA is very compact
  - Comparing with CS-KWLM approximation approach
- The grammar size of the CS-KWLM interpolated LM is much larger
  - Due to additional history states derived from the keyword list (CS-KWLM)

<table>
<thead>
<tr>
<th>Vietnamese grammars</th>
<th># arcs</th>
<th># states</th>
<th>File Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>n-gram baseline</td>
<td>38,713</td>
<td>17,616</td>
<td>812 Kb</td>
</tr>
<tr>
<td>KW-aware grammar (global $\kappa = 0.00005$)</td>
<td>66,913</td>
<td>24,215</td>
<td>1.3 Mb</td>
</tr>
<tr>
<td>CS-KWLM Int ($\alpha = 0.6$)</td>
<td>381,461</td>
<td>165,063</td>
<td>7.8 Mb</td>
</tr>
</tbody>
</table>
Vietnamese System Performance

- KW-aware framework is very effective
  - Reduced 1/3 #miss
  - Achieved 57% relative ATWV improvement
- Both realizations have a similar overall performance
  - Though the file size of the KW-aware grammar is much smaller

<table>
<thead>
<tr>
<th>Vietnamese [evalpart 1]</th>
<th># Miss</th>
<th>ATWV</th>
</tr>
</thead>
<tbody>
<tr>
<td>n-gram baseline</td>
<td>2,562</td>
<td>0.2093</td>
</tr>
<tr>
<td>KW-aware framework</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KW-aware grammar (global $\kappa = 0.00005$)</td>
<td>1,589</td>
<td>0.3224</td>
</tr>
<tr>
<td>CS-KWLM Interpolation ($\alpha=0.6$)</td>
<td>1,651</td>
<td>0.3287</td>
</tr>
</tbody>
</table>
Tamil System Performance

- Similar observation could be found in the Tamil task
  - The effect of KW-aware framework is language independent
  - 24% reduction on #miss
  - ~50% relative ATWV improvement

<table>
<thead>
<tr>
<th>Tamil [evalpart 1]</th>
<th># Miss</th>
<th>ATWV</th>
</tr>
</thead>
<tbody>
<tr>
<td>n-gram baseline</td>
<td>3,663</td>
<td>0.2128</td>
</tr>
<tr>
<td>KW-aware framework</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KW-aware grammar (global $\kappa = 0.0000347$)</td>
<td>2,830</td>
<td>0.3102</td>
</tr>
<tr>
<td>CS-KWLM Interpolation ($\alpha=0.3$)</td>
<td>2,689</td>
<td>0.3160</td>
</tr>
</tbody>
</table>
ATWV analysis for keywords of different length

- *n*-gram LM baseline
  - Suffers from KW prior underestimation
  - Especially for multi-word keywords
- KW-aware grammar works better on long keywords
  - L > 2
  - Standalone keyword path
    - Boost the probability of keyword sequence only

Diagram:
- Tamil Task
- X-axis: Keyword Length L (#word)
- Y-axis: ATWV
- Lines:
  - Blue: n-gram baseline
  - Red: CSKWLM Int
  - Green: KW-aware grammar
Conclusion

- Exact realization of the KW-aware grammar
  - Using WFSA with disambiguation symbols
  - The improvement of the KW-aware framework over the \(n\)-gram baseline is consistent across languages
- Two realizations available for the KW-aware framework
  - Both alleviate the keyword prior underestimation problem
    - Achieve similar performance enhancement
    - Exact realization – insert standalone KW paths
      - Much more compact in final grammar WFSA
      - Better performance for long keywords (L>2)
    - Approximation – by CS-KWLM interpolation
      - Very easy to be realized
      - Better performance for short keywords (L \(\leq\) 2)
      - Can be applied to all LVCSR-based KWS system
  - Suit different scenarios
  - Further system combination may achieve better KWS results
Thank You!!