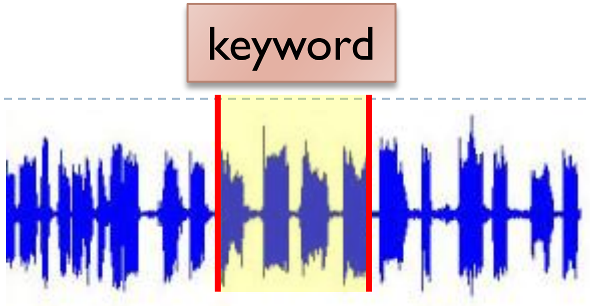
Georgia Institute of Technology

A*STAR Institute for Infocomm Research

# 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

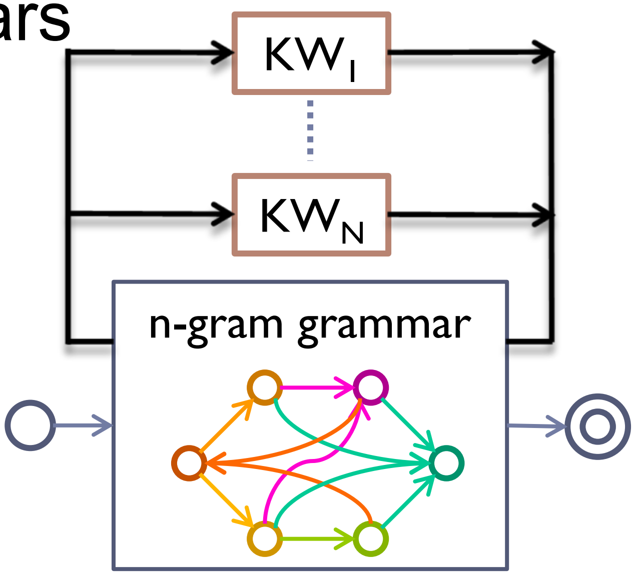
# Introduction



- 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

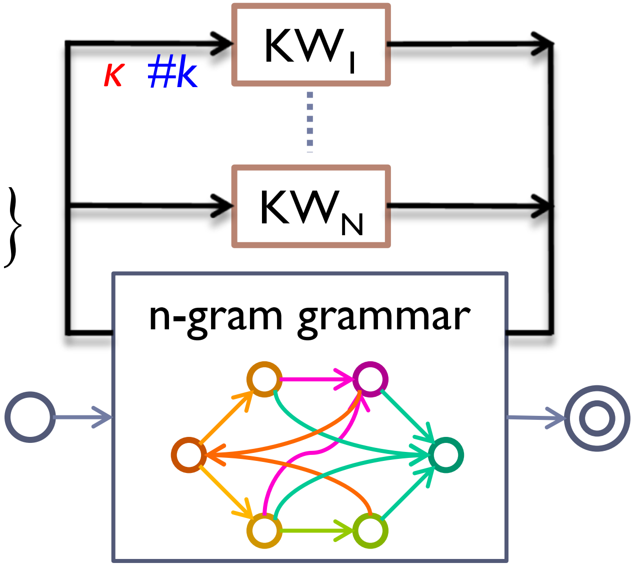
- Integrate keyword-filler loop grammars into LVCSR-based KWS [1]
  - Boost KW probabilities in the decoding grammar
- Approximate the boosting effect
  - Context-Simulated keyword language model (CS-KWLM) interpolated LMs
    - Train a context-simulated KWLM using the system keyword list
    - Interpolate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



[1] I.-F Chen, C. Ni, et al., "A Novel Keyword+LVCSR-Filler Based Grammar Network Representation for Spoken Keyword Search, " in *Proc. ISCLSP* 2014

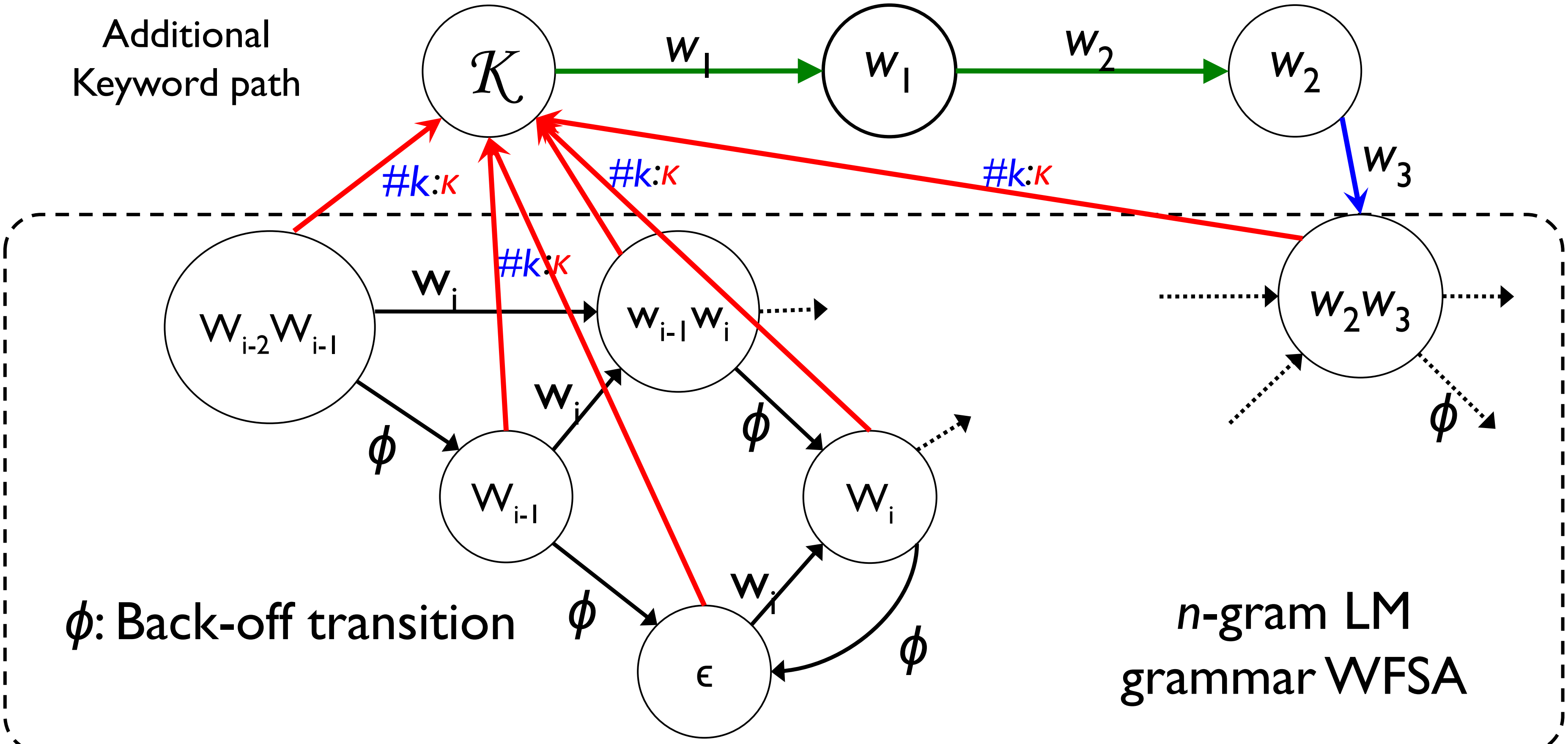
# Exact Realization of the KW-aware Grammar

- Weighted Finite State Automata (WFSA)
  - Insert KW paths to the *n*-gram LM grammar WFSA of the system
  - For each *kw*=$w_1 \ldots w_L$
  - $P_{KW-aware}(kw \mid h) = \max\left\{ P_{n-gram}(kw \mid h)\ ,\ \kappa \right\}$
    - $P_{n-gram}(kw \mid h) = \prod_{i=1}^{L} P(w_i \mid h_i)$
    - $\kappa$ : 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 $w_1 w_2 w_3$
- 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)

| Vietnamese grammars | # arcs | # states | File Size |
|---|---|---|---|
| *n*-gram baseline | 38,713 | 17,616 | 812 Kb |
| KW-aware grammar<br>(global $\kappa = 0.00005$ ) | 66,913 | 24,215 | 1.3 Mb |
| CS-KWLM Int ($\alpha = 0.6$) | 381,461 | 165,063 | 7.8 Mb |

# 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

| Vietnamese [evalpart 1] | | # Miss | ATWV |
|---|---|---|---|
| n-gram baseline | | 2,562 | 0.2093 |
| KW-aware framework | KW-aware grammar<br>(global $\kappa$ = 0.00005 ) | **1,589** | 0.3224 |
| | CS-KWLM Interpolation<br>($\alpha$=0.6) | 1,651 | **0.3287** |

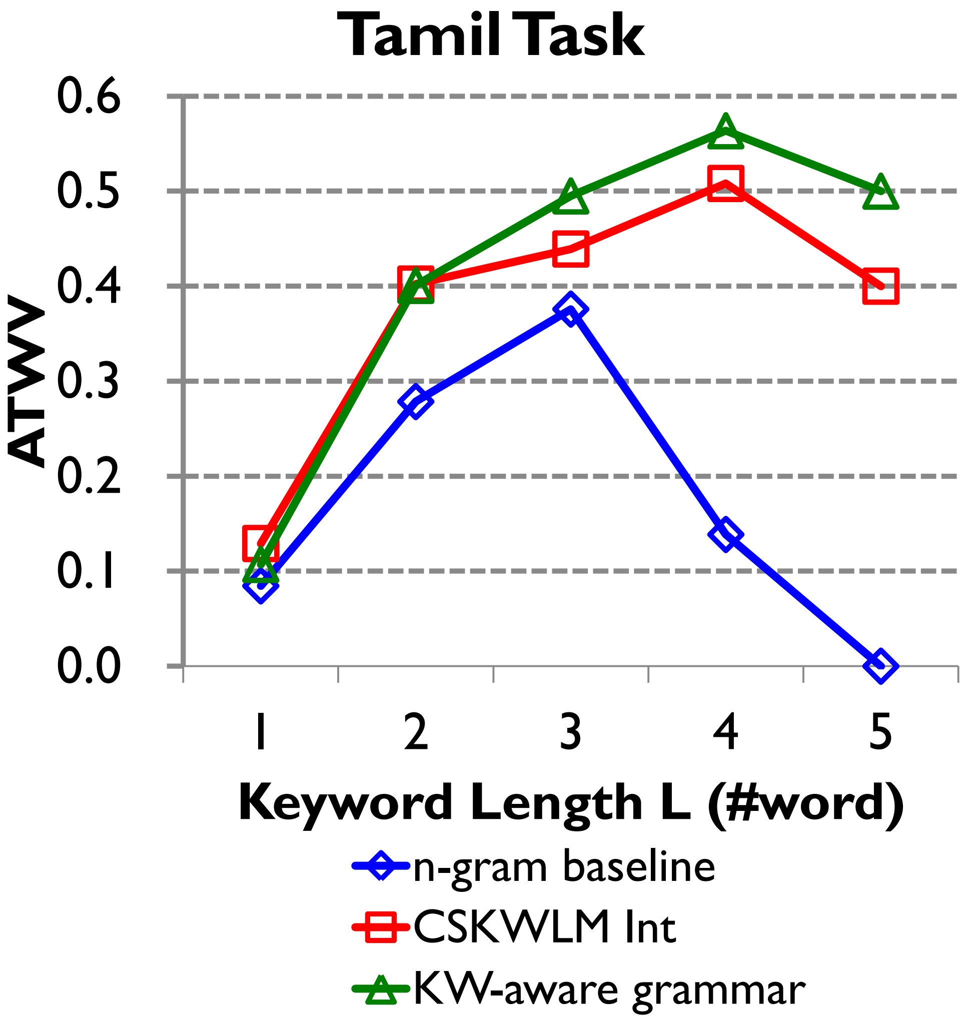
# Tamil System Performance

- 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

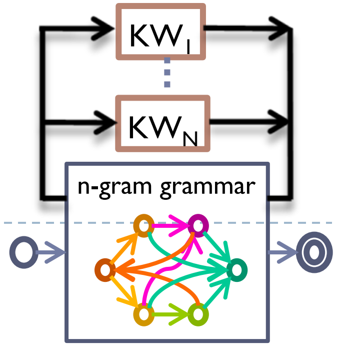
| Tamil [evalpart 1] | | # Miss | ATWV |
|---|---|---|---|
| n-gram baseline | | 3,663 | 0.2128 |
| KW-aware framework | KW-aware grammar<br>(global $\kappa$ = 0.0000347 ) | 2,830 | 0.3102 |
| | CS-KWLM Interpolation<br>($\alpha$=0.3) | **2,689** | **0.3160** |

# 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



# 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 <= 2)
    - Can be applied to all LVCSR-based KWS system
  - Suit different scenarios
  - Further system combination may achieve better KWS results

Thank You!!