EECS E6870: Lecture 12: Special Topics – Spoken Term Detection

Stanley F. Chen, Michael A. Picheny and Bhuvana Ramabhadran
IBM T. J. Watson Research Center
Yorktown Heights, NY 10549

stanchen@us.ibm.com, picheny@us.ibm.com,
bhuvana@us.ibm.com

December 1, 2009
What is it?

- Search for specific terms in large amount of speech content (key word spotting)
- Enable open vocabulary search
- Applications:
  - Call monitoring
  - Market intelligence gathering
  - Customer analytics
  - On-line media search
Something like this........

ALJZ Sep. 15, 2005 5:48 AM EDT
Local Broadcast Time: 5:48 AM EDT

being away from the battle field makes it hard to predict who is right and who is wrong, but I think they should be involved in building thier country and writing the constitution and the whole political movement in Iraq reports, so do you think its better to be involved in the peaceful movement? yes this is beneficial
being away from the battlefield makes it hard to predict who is right and who is wrong, but I think they should be involved in building their country and writing the constitution and the whole political movement in Iraq reporter: so do you think it's better to be involved in the peaceful movement? yes, this is beneficial.

what do you think about the conflict of cultures? I think that there are some people who try to overspeak about this issue but let me tell you that Islam is against that idea and Islam is an open-minded religion and it tolerates the difference.
Historically……

• Keyword spotting (KWS)
  • In the 90s....
    • Use of filler models (parallel set of phone HMMs)
    • Likelihood ratio comparisons
    • Phone lattices for spoken document retrieval
  • Two step approach
    • Coarse step: identify candidate regions quickly
    • Detailed step: Better models to zero in on region of interest

• Phone decoding and its various flavors
• LVCSR
Historically……

• Unreliable transcriptions: high error rate in one best transcripts
  • Search on lattices and/or confusion networks (CN)

• Efficient indexing and search algorithms
  • General Indexation of Weighted Automata [Saraclar 2004, Allauzen et al., 2004]
  • Posting list [JURU/Lucene] [Carmel et al. 2001, Mamou et al. 2007]

• Out Of Vocabulary queries: information bearing words
  • OOV pronunciation modeling [Can et al. 2009, Cooper, et al, 2009]
  • Search on subword decoding [Saraclar and Sproat 2004, Mamou et al, 2007, Chaudhari and Picheny, 2007]
Out of Vocabulary Terms

- ASR vocabulary might not cover all words of interest
  - Information bearing words
  - Loss of context impacts word error rate
  - Special interest for spoken term retrieval
- Challenges in OOV detection and recovery
  - Rare foreign terms with a diverse set of pronunciations
  - Confusability with similar sounding in-vocabulary term
  - Language model information is missing
Representing and detecting OOV terms

- **Use a combination of word and subword units:**
  - Identify set of words and subword units (fragments) for good coverage
  - Represent LM text as a combination of words and fragments
  - Build a Hybrid Language Model and Lexicon
  - Acoustic models for hybrid system are the same as word-based LVCSR system

- **Example:**
  - `<s> THE WORKS OF ZIYAD HAMDI WERE RECENTLY AUCTIONED</s>`
  - `<s> THE WORKS OF Z_IY Y_AE_D HH_AE_M D_IY WERE RECENTLY AUCTIONED </s>`
Spoken Term Detection

Indexing

Speech Database

Retrieval System

Word Index

Phonetic Index

query

Preprocess

retrieve

> T?

yes

no

ignore

Search
What speech Recognition output structures do we index?

- 1-best: I HAVE IT VEAL FINE
- Lattice:

Word Confusion networks (WCN):

```
I     HAVE     IT     VEAL     FINE
  MOVE     IT     VERY     OFTEN
      MOVE
      SIL
```
Evaluation Metrics

- The basic idea is to count misses and false alarms for each query and to average this number across all queries
  - F-measure: Trade-off between Precision and Recall
  - Number of False Alarms per hour
  - In a task like distillation in GALE, false alarms may not matter as long as the first page of results contains at least an entry on what you are looking for...
  - Average Term Weighted Value: Weighted average of misses and false alarms
Indexing Architectures

- **JURU/Lucene**:
  - Extension of information retrieval methods for text (text-based search engine)
  - Use posting lists to store time, probabilities and index units
  - Compact representation but not very flexible

- **Transducer based**:
  - Represent indices as transducers
  - More flexible at the cost of compactness
What can you do with an FST-based indexing system?

- Allows us to search for complex regular expressions

[healthcare 0.6, health care 0.4] [reform 0.8, plan 0.2]

- Easy to do fuzzy matching

- We can search using audio snippets: query-by-example (QbyE)
NIST Spoken Term Detection Evaluation

- **Detection Task**
  - Count misses and false alarms for each query
  - Average across all queries

- **Actual Term-Weighted Value (ATWV)**

\[
ATWV = 1 - \frac{1}{N_{terms}} \sum_{t \in terms} (P_{miss}(t) + \beta \cdot P_{fa}(t))
\]

\[
P_{miss}(t) = 1 - \frac{N_{corr}(t)}{N_{true}(t)} \quad P_{fa}(t) = \frac{N_{spurious}(t)}{Total - N_{true}(t)}.
\]

B=1000, False alarms are heavily penalized

- **Broadcast News**
- **Telephone Speech**
- **Conference Meetings**
Actual Term Weighted Value [NIST STD 2006 Evaluation Plan]:

\[
ATWV = 1 - \frac{1}{Q} \sum_{q=1}^{Q} P_{\text{miss}}(q) + \beta P_{FA}(q)
\]

\[
P_{\text{miss}}(q) = 1 - \frac{N_{\text{corr}}(q)}{N_{\text{true}}(q)} \\
P_{FA}(q) = \frac{N_{\text{spurious}}}{T - N_{\text{true}}(q)}
\]

\[
Q \quad = \quad \text{number of queries} \\
N_{\text{true}} \quad = \quad \text{occurrences in reference} \\
N_{\text{spurious}} \quad = \quad \text{spurious instances retrieved} \\
N_{\text{corr}} \quad = \quad \text{correctly retrieved instances} \\
\beta \quad = \quad \text{user defined parameter, in STD 06 Eval } \beta = 999.99 \\
T \quad = \quad \text{seconds of audio (secs)}
\]
Word-Fragment Hybrid systems

- Posterior probability of fragments in a given region is a good indicator of presence of OOVs.
- Hybrid systems represent OOV terms better in phonetic sense than pure word systems or pure phonetic systems.
OOV Detection with hybrid systems
NIST 2006 Evaluation (English)

<table>
<thead>
<tr>
<th>system</th>
<th>BN</th>
<th>CTS</th>
<th>CONFMTG</th>
</tr>
</thead>
<tbody>
<tr>
<td>TWV Dry-Run P</td>
<td>0.8498</td>
<td>0.6597</td>
<td>0.2921</td>
</tr>
<tr>
<td>ATWV Eval P</td>
<td>0.8485</td>
<td>0.7392</td>
<td>0.2365</td>
</tr>
<tr>
<td>MTWV Eval P</td>
<td>0.8532</td>
<td>0.7408</td>
<td>0.2508</td>
</tr>
<tr>
<td>ATWV Eval C1</td>
<td>0.8485</td>
<td>0.7392</td>
<td>0.0016</td>
</tr>
<tr>
<td>MTWV Eval C1</td>
<td>0.8532</td>
<td>0.7408</td>
<td>0.0115</td>
</tr>
<tr>
<td>ATWV Eval C2</td>
<td>0.8293</td>
<td>0.6763</td>
<td>0.1092</td>
</tr>
<tr>
<td>MTWV Eval C2</td>
<td>0.8293</td>
<td>0.6763</td>
<td>0.1092</td>
</tr>
<tr>
<td>ATWV Eval C3</td>
<td>0.8279</td>
<td>0.7101</td>
<td>0.2381</td>
</tr>
<tr>
<td>MTWV Eval C3</td>
<td>0.8319</td>
<td>0.7117</td>
<td>0.2514</td>
</tr>
</tbody>
</table>

- Retrieval performances are improved using WCNs, relatively to 1-best path.
- Our ATWV is close to the MTWV; we have used appropriate thresholds for pruning bad results.
WFST-based indexing

Recipe: preprocess lattices, build index, search

Preprocess:

(1)

(2)
WFST-based indexing

Recipe: preprocess lattices, build index, search

- Preprocess:

(1) Preprocess lattices:

0 \[ \rightarrow \begin{array}{c}
 b \uparrow 0.10536 \\
 a \downarrow 2.3026
\end{array} \rightarrow
1 \[ \rightarrow \begin{array}{c}
 a \uparrow 0.35667 \\
 b \downarrow 1.6094 \\
 c \downarrow 2.3026
\end{array} \rightarrow
2 \[ \rightarrow
3

Include time-information

(2) Preprocess lattices:

0 \[ \rightarrow
1 \[ \rightarrow
2 \[ \rightarrow
3

Spoken Term Detection
An Example: preprocess

Recipe: preprocess lattices, build index, search

- **Preprocess:**

1. Include time-information

2. Normalize weights
WFST-based indexing: an example

(1)
WFST-based indexing: an example

set output labels to “eps”
WFST-based indexing: an example

(1) add new start state and new end state
WFST-based indexing: an example

(1) Add arc from 4 to each state S in original machine. Weight is shortest distance in log semiring between state S to BLUE state.
WFST-based indexing: an example

(1) Add arc from 4 to each state S in original machine. Weight is shortest distance in log semiring between state S to BLUE state.
WFST-based indexing: an example

Add arc from 4 to each state $S$ in original machine. Weight is shortest distance in log semiring between state $S$ to **BLUE** state
WFST-based indexing: an example

(1) Add arc from each state S in original machine to state 5. Weight is shortest distance in log semiring between state S to RED state.
for each query in query-list
- compile query into string fst
  - compose query with index fst to get utt-ids
  - padfst = pad query fst on left and right
  - for each utt-id
    - load utt-fst
    - shortest-path(compose(padded-query, utt-fst))
    - read off output labels of marked arcs
Augmenting STD with web based pronunciations

- Generating pronunciations for OOV terms is important for spoken term detection
- The internet can serve as a gigantic pronunciation corpus
- Work done as part of CLSP 2008 workshop
- Find pronunciations derived from the web:
  - IPA Pronunciations: Uses International Phonetic Alphabet:
    • Lorraine Albright /ɔɪ braɪt/ (Wikipedia)
  - Ad-hoc Pronunciations: Uses informal pronunciation:
    • Bruschetta (pronounced broo-SKET-uh)
    • Bazell (pronounced BRA-zell by the lisping Brokaw)
    • Ahmadinijad (pronounced "a mad dog on Jihad")
- Normalize, filter and refine web-pronunciations (esp. AdHoc)
Utility of web-pronunciations (from JHU workshop ’08)

<table>
<thead>
<tr>
<th>Better</th>
<th>Pronunciations</th>
<th>Ref/Corr/FA/Miss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>L2S</td>
<td>Web Based</td>
</tr>
<tr>
<td>ALBRIGHT</td>
<td>aelbrayt</td>
<td>aolbrayt</td>
</tr>
<tr>
<td>GREENSPAN</td>
<td>griynspaan</td>
<td>griynspaan</td>
</tr>
<tr>
<td>SHIMON</td>
<td>shihmaxn</td>
<td>shihmown</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Worse</th>
<th>Pronunciations</th>
<th>Ref/Corr/FA/Miss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>L2S</td>
<td>Web Based</td>
</tr>
<tr>
<td>FREUND</td>
<td>froynd</td>
<td>frehnd</td>
</tr>
<tr>
<td>SANTO</td>
<td>saentow</td>
<td>sax/ey/ax/ehnt</td>
</tr>
<tr>
<td>THIERRY</td>
<td>thiyaaxriy</td>
<td>tehriy</td>
</tr>
</tbody>
</table>

Names resemble portions of common words and prefix/suffixes
Large number of false alarms
THIERRY :: -TARY :: MILLITARY, VOLUNTARY
Experiments/Data

OOVCORP [JHU Workshop]

Test-set:
- 100 Hour
- 1290 OOV queries (min 5 instances/word)
- All queries larger than 4 phones.

Training set (word system):
- 300 Hours SAT system
- 400M words, vocabulary: 83K
- WER on RT04 BN: 19.4%

Hybrid system:
- Lexicon: 81.7K words and 20K fragments

DEV06

Test-set:
- Development set used for NIST STD 2006 Evaluation
- 3 Hour BN
- 1107 queries, 16 OOVs

Training set:
- IBM BN system
- vocabulary: 84K
## Results

### DEV06

<table>
<thead>
<tr>
<th>Data</th>
<th>P(FA)</th>
<th>P(miss)</th>
<th>ATWV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word lattices (index:word, query:word)</td>
<td>0.00008</td>
<td>0.134</td>
<td>0.7991</td>
</tr>
<tr>
<td>Word CNs (index:word, query:word)</td>
<td>0.00007</td>
<td>0.094</td>
<td>0.8459</td>
</tr>
<tr>
<td>Hybrid lattices (index:phonetic, query:phones)</td>
<td>0.00008</td>
<td>0.240</td>
<td>0.6779</td>
</tr>
<tr>
<td>Merged (IV:word, OOV:phones)</td>
<td>0.00007</td>
<td>0.093</td>
<td>0.8490</td>
</tr>
</tbody>
</table>

### OOVCORP (OOV-only queries, phonetic index)

<table>
<thead>
<tr>
<th>Data</th>
<th>Pron model</th>
<th># Best</th>
<th>P(FA)</th>
<th>P(miss)</th>
<th>ATWV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word Lat</td>
<td>reflex</td>
<td>N/A</td>
<td>0.00004</td>
<td>0.638</td>
<td>0.325</td>
</tr>
<tr>
<td>Hybrid Lat</td>
<td>reflex</td>
<td>N/A</td>
<td>0.00002</td>
<td>0.639</td>
<td>0.342</td>
</tr>
<tr>
<td>Word Lat</td>
<td>L2S-weighted</td>
<td>6</td>
<td>0.00002</td>
<td>0.674</td>
<td>0.305</td>
</tr>
<tr>
<td>Hybrid Lat</td>
<td>L2S-weighted</td>
<td>6</td>
<td>0.00002</td>
<td>0.630</td>
<td>0.342</td>
</tr>
</tbody>
</table>
## Results

### DEV06

<table>
<thead>
<tr>
<th>Data</th>
<th>P(FA)</th>
<th>P(miss)</th>
<th>ATWV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word lattices (index:word, query:word)</td>
<td>0.00008</td>
<td>0.134</td>
<td>0.7991</td>
</tr>
<tr>
<td>Word CNs</td>
<td>0.00007</td>
<td>0.094</td>
<td>0.8459</td>
</tr>
<tr>
<td>Hybrid lattices (index:phonetic, query:phones)</td>
<td>0.00008</td>
<td>0.240</td>
<td>0.6779</td>
</tr>
<tr>
<td>Merged (IV:word, OOV:phones)</td>
<td>0.00007</td>
<td>0.093</td>
<td>0.8490</td>
</tr>
</tbody>
</table>

### OOVCORP (OOV-only queries, phonetic index)

<table>
<thead>
<tr>
<th>Data</th>
<th>Pron model</th>
<th># Best</th>
<th>P(FA)</th>
<th>P(miss)</th>
<th>ATWV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word Lat</td>
<td>reflex</td>
<td>N/A</td>
<td>0.00004</td>
<td>0.638</td>
<td>0.325</td>
</tr>
<tr>
<td>Hybrid Lat</td>
<td>reflex</td>
<td>N/A</td>
<td>0.00002</td>
<td>0.639</td>
<td>0.342</td>
</tr>
<tr>
<td>Word Lat</td>
<td>L2S-weighted</td>
<td>6</td>
<td>0.00002</td>
<td>0.674</td>
<td>0.305</td>
</tr>
<tr>
<td>Hybrid Lat</td>
<td>L2S-weighted</td>
<td>6</td>
<td>0.00002</td>
<td>0.630</td>
<td>0.342</td>
</tr>
</tbody>
</table>

| True Instances | 23322 |
| Hits           | 8105  |
| False Alarms   | 10446 |
FST-based STD vs JURU/Lucene

<table>
<thead>
<tr>
<th>WFST-based</th>
<th>JURU-based 2006 system</th>
</tr>
</thead>
<tbody>
<tr>
<td>lattice and confusion networks</td>
<td>confusion networks</td>
</tr>
<tr>
<td>no boosting</td>
<td>boost posteriors based on ranking</td>
</tr>
<tr>
<td>no query-length normalization</td>
<td>query-length normalization</td>
</tr>
<tr>
<td>term specific threshold</td>
<td>global threshold</td>
</tr>
</tbody>
</table>

WFST-based vs JURU-based

<table>
<thead>
<tr>
<th>System</th>
<th>Data</th>
<th>P(FA)</th>
<th>P(miss)</th>
<th>MTWV</th>
<th>ATWV</th>
</tr>
</thead>
<tbody>
<tr>
<td>JURU 2006</td>
<td>Word &amp; Phonetic CNs</td>
<td>0.00005</td>
<td>0.108</td>
<td>0.8379</td>
<td>0.8348</td>
</tr>
<tr>
<td>WFST-based</td>
<td>Word CNs &amp; Phonetic Lats</td>
<td>0.00007</td>
<td>0.093</td>
<td>0.8392</td>
<td><strong>0.8490</strong></td>
</tr>
</tbody>
</table>
Increasing Hits

- **Increasing hits # 1: include phonetic confusability in query**
  - Create phone-to-phone confusability matrix.
  - Model phonetic confusability using posteriors of NN-based acoustic model and aligned reference [Upendra 2009].
  - Easy to incorporate in the WFST-based framework
Increasing Hits

- **Increasing hits # 1: include phonetic confusability in query**
  
  - Create phone-to-phone confusability matrix.
  
  - Model phonetic confusability using posteriors of NN-based acoustic model and aligned reference [Upendra 2009].
  
  - Easy to incorporate in the WFST-based framework
Increasing Hits

- **Increasing hits # 1: include phonetic confusability in query**
  - Create phone-to-phone confusability matrix.
  - Model phonetic confusability using posteriors of NN-based acoustic model and aligned reference [Upendra 2009].
  - Easy to incorporate in the WFST-based framework

\[
\text{oldq} = \text{shortest-pathN}(q \circ L2S)
\]

\[
\text{newq} = \text{shortest-pathN}(q \circ L2S \circ P2P)
\]
Reducing False Alarms

- **Reducing FAs #1**: Query-length normalization [Mamou et al. 2007]:

\[
\text{score}(q, \text{hit}, \gamma) = p(\text{hit})^{\frac{\gamma}{\text{avg.-duration}(q)}}, \quad \gamma \in [0, 1]
\]

- **Reducing FAs #2**: OOV-detection [Arastrow et al. 2009]

  – Simplest OOV detector: use posterior probabilities of fragments in a confusion bin (hybrid CN) as indicator of OOV region [frag_p > 0]

  – Reduce confidence of hit if query and region do not match.
### Experiments: OOVCORP

- **Increasing hits: Phone-to-Phone transducer**

<table>
<thead>
<tr>
<th></th>
<th>none</th>
<th>P2P-10best</th>
<th>P2P-20best</th>
<th>P2P-100best</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATWV</td>
<td>0.342</td>
<td>0.368</td>
<td>0.383</td>
<td>0.3904</td>
</tr>
<tr>
<td>% rel improv</td>
<td>-</td>
<td>7.6%</td>
<td>12%</td>
<td>15.9%</td>
</tr>
</tbody>
</table>
Experiments: OOVCORP

- Increasing hits: Phone-to-Phone transducer

<table>
<thead>
<tr>
<th></th>
<th>none</th>
<th>P2P-10best</th>
<th>P2P-20best</th>
<th>P2P-100best</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATWV % rel improv</td>
<td>0.342</td>
<td>0.368</td>
<td>0.383</td>
<td>0.3904</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>7.6%</td>
<td>12%</td>
<td>15.9%</td>
</tr>
</tbody>
</table>

- OOV-detection + length-normalization + cache: pron-model: P2P-20best

<table>
<thead>
<tr>
<th>OOV-det</th>
<th>γ-norm</th>
<th>cache</th>
<th>P(FA)</th>
<th>P(miss)</th>
<th>ATWV</th>
<th>improv</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td></td>
<td>0.00004</td>
<td>0.575</td>
<td>0.383</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td></td>
<td>0.00005</td>
<td>0.555</td>
<td>0.394</td>
<td>0.2%</td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>0.00005</td>
<td>0.555</td>
<td>0.394</td>
<td>2.87%</td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>0.00006</td>
<td>0.557</td>
<td>0.383</td>
<td>0%</td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>0.00004</td>
<td>0.551</td>
<td>0.405</td>
<td>18.4%</td>
<td></td>
</tr>
</tbody>
</table>
Query-by-Example (QbyE)

- Spoken Term Detection when the terms of interest are acoustic examples: Query by Example (QbyE).
  - User identifies region of interest in speech stream and requests for similar examples.
  - User speaks query: speech to speech retrieval.
- Focus on improving performance for Out Of Vocabulary (OOV) words.
- Demonstrates flexibility of FST-based indexing system
Query Generation for QbyE

- **Lattice Cuts**: User selects a region of interest in the audio stream
  - Represent region of interest by excising lattice corresponding to the decode for the region
  - Query representation generated by the same ASR system which generates the index

- **Isolated decodes**: User presents example of audio
  - Use lattice from an isolated decode of the audio example

- **The queries for both cases are graph structures similar to ASR lattices**

- **Pruned representation of queries found to be faster, more robust and generate lower false alarms**
Query by Example: Key results

- QbyE typically perform significantly better than textual queries for OOV terms (about 20% relative in ATWV).
- Queries represented as *lattice-cuts* from the lattices of interest yield better STD performance than *isolated-decode* queries.
- Addressing FA rates associated with multi-path queries improves performance significantly.
- QbyE can enhance performance of textual queries when using a two-pass approach.