Improved Modeling of Cross-Decoder Phone Co-occurrences in SVM-based Phonotactic Language Recognition

Mikel Penagarikano, Amparo Varona, Luis J. Rodríguez-Fuentes, Germán Bordel

Software Technologies Working Group (http://gtts.ehu.es)
Department of Electricity and Electronics, University of the Basque Country
Barrio Sarriena s/n, 48940 Leioa, Spain
email: mikel.penagarikano@ehu.es

Odyssey 2010, Brno, Czech Republic
July 1, 2010
Outline

1. Introduction
2. Baseline SVM-based Phonotactic System
3. Cross-Decoder Phone Co-occurrences based System
4. Experimental Setup
5. Results
6. Summary
Motivation

- Most common approaches to phonotactic language recognition deal with several independent phone decodings.

- These decodings are processed and scored in a fully uncoupled way and no cross-decoder dependencies are exploited for language modeling, information being fused only at the score level.

- Certain sounds from languages not covered by (not matching) the decoders may be better represented by cross-decoder outputs.
Cross-stream (cross-decoder) information previously applied for speaker recognition in the Johns Hopkins University (JHU) 2002 Workshop, where two decoupled time and cross-stream systems were integrated at the score level.

Cross-stream (cross-decoder) information previously applied for speaker recognition in the Johns Hopkins University (JHU) 2002 Workshop, where two decoupled time and cross-stream systems were integrated at the score level.


Some years later, cross-stream dependencies were also used via multi-string alignments in a language recognition application

Common approach to phonotactic language recognition:

\[ N \text{ Phone Decoders} + L \text{ SVM-based Language Models} + \text{Gaussian Backend} + \text{Linear Fusion} \]
Exploit cross-decoder dependencies using time-synchronous (frame level) phone co-occurrences.

In a two decoder scenario:

- In a $D$-decoder scenario:
  - Build a single $D$-phone co-occurrence system
  - Build $D!/k!(D - k)!$ $k$-phone co-occurrence systems
Approach 1: n-grams of phone co-occurrences

- First introduced in Penagarikano et al., ICASSP 2010.
- Get a frame-synchronous sequence of multi-phone (\(k\)-phone co-occurrence)
- Two type of sequence segments can be identified
  - *Stationary segments*: relatively long portions of speech for which decoders keep the same labels
  - *Transitional segments*: mainly appearing at phone borders (cross-decoder desynchronization)
- Transitional segments are removed and stationary segments are collapsed.
Approach 1: n-grams of phone co-occurrences

![Diagram showing time and phone co-occurrences across decoders]

**DECODER 1**
- a a a a a a
- c c c c c c
- b b b b b b

**DECODER 2**
- x x x x x x
- y y y y y y
- z z z z z z

**DECODER 3**
- n n n n n n
- p p p p p p
- k k k k k k

**Filtering**
- a_x
- a_y
- c_y
- b_z

**Reduction**
- a_x_n
- a_y_n
- c_y_p
- b_z_k
Approach 1: n-grams of phone co-occurrences

- Standard phonotactic approach is performed on the resulting $k$-phone sequence.

- ... not so standard
  - Number of different $k$-phones (1-grams): 2500 ($k = 2$), 124000 ($k = 3$)
  - The number of n-grams increases exponentially.
  - A full bag of n-grams strategy is infeasible.

- Only the most frequent n-gram counts are included in the supervector.
Approach 2: co-occurrences of phone n-grams

- In the previous approach, cross-decoder desynchronization affects the time modeling (n-grams).

- Exploit cross-decoder dependencies using time-synchronous (frame level) phone n-gram co-occurrences.

- Directly compute the n-gram co-occurrence counts from the decodings.
  - Each phone n-gram is counted once for each decoder, so its count is distributed among all the frames it spans.
  - The contribution corresponding to a given phone n-gram at a given frame is distributed among all the co-occurrences.
  - The sum of the counts of phone n-grams co-occurrences is equal to the average number of n-grams.

- Only the most frequent co-occurrence counts are included in the supervector.
Approach 2: co-occurrences of phone n-grams

\[
\begin{align*}
\text{count}(c_{-y}) &= \frac{1}{2} \left( \frac{1}{8.1} + \frac{1}{13.1} \right) \\
\text{count}(ac_{-xy}) &= \frac{1}{2} \left( \frac{1}{17.2} + \frac{1}{19.2} \right) \\
\text{count}(ac_{-yz}) &= \frac{1}{2} \left( \frac{1}{17.2} + \frac{1}{18.2} \right) \\
\text{count}(cb_{-xy}) &= \frac{1}{2} \left( \frac{1}{15.2} + \frac{1}{19.2} \right) \\
\text{count}(cb_{-yz}) &= \frac{1}{2} \left( \frac{1}{15.2} + \frac{1}{18.2} \right)
\end{align*}
\]
Training, development and test corpora

- Limited to those distributed by NIST to all LRE2007 participants
  - Call-Friend Corpus
  - OHSU Corpus provided by NIST for LRE05
  - Development corpus provided by NIST for LRE07

- 10 conversations per language randomly selected for development purposes.

- Each development conversation was further split in segments containing 30 seconds of speech.

- Evaluation was carried out on the LRE07 evaluation corpus, specifically on the 30-second, closed-set condition.
Evaluation measures

- Most usual performance measures used in language recognition systems.
  - $DET$ plots & $EER$: not providing calibration information.
  - $C_{avg}$ & $C_{min}$: application dependent costs.
Evaluation measures

- Most usual performance measures used in language recognition systems.
  - *DET* plots & *EER*: not providing calibration information.
  - *C*_{avg} & *C*_{min}: application dependent costs.

- We prefer *C*_{llr} (more precisely, *C*_{mxe})
  - It is used as an alternative performance measure in NIST evaluations.
  - It evaluates the application independent system performance by means of a single numerical value (and appealing units: **bits**).
  - \( \Delta = \log_2 N - C_{mxe} \) gives the effective amount of information that the recognizer delivers to the user, given no prior information.
  - The lower *C*_{mxe} is, the more informative our system is.

*DET* plots, *EER* and detection cost are kept.
Evaluation measures

- Most usual performance measures used in language recognition systems.
  - \(DET\) plots & \(EER\): not providing calibration information.
  - \(C_{avg}\) & \(C_{min}\): application dependent costs.

- We prefer \(C_{llr}\) (more precisely, \(C_{mxe}\))
  - It is used as an alternative performance measure in NIST evaluations.
  - It evaluates the application independent system performance by means of a single numerical value (and appealing units: \textbf{bits}).
  - \(\Delta = \log_2 N - C_{mxe}\) gives the effective amount of information that the recognizer delivers to the user, given no prior information.
  - The lower \(C_{mxe}\) is, the more informative our system is.

- However, we keep \(DET\) plots, \(EER\) and detection cost.
Freely available software was used in all the stages

- **Phone Decoders:** The TRAPS/NN phone decoders developed by the Brno University of Technology (BUT) for Czech (CZ), Hungarian (HU) and Russian (RU).

- **SVM modeling:** *LIBLINEAR* (a fast linear-only version of libSVM). Modified by adding some lines of code to get the regression values (instead of class labels).

- **Gaussian Backend & Fusion:** *FoCal Multi-class* toolkit by Niko Brummer.
Configuration

**BUT TRAPS/NN CZ, HU & RU phone decoders**

- Before doing phone tokenization, an energy-based voice activity detector is applied to split and remove non-speech segments.
- Non phonetic units (int, pau and spk) are mapped to silence (sil).
- Number of resulting phonemes: 43 (CZ), 59 (HU) and 49 (RU).
- 1-best decoding.
Configuration

**BUT TRAPS/NN CZ, HU & RU phone decoders**
- Before doing phone tokenization, an energy-based voice activity detector is applied to split and remove non-speech segments.
- Non phonetic units (int, pau and spk) are mapped to silence (sil).
- Number of resulting phonemes: 43 (CZ), 59 (HU) and 49 (RU).
- 1-best decoding.

**LIBLINEAR**
- Phone sequences are modelled by means of Support Vector Machines
- SVM vectors consist of counts of phone n-grams (up to trigrams), converted to frequencies and weighted with regard to their background probabilities as $w_i = \min \left( C, \frac{1}{\sqrt{p(d_i|\text{background})}} \right)$, with $C = 300$
<table>
<thead>
<tr>
<th></th>
<th>EER</th>
<th>$C_{LLR}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baseline</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZ</td>
<td>5.67%</td>
<td>0.8259</td>
</tr>
<tr>
<td>HU</td>
<td>5.10%</td>
<td>0.7434</td>
</tr>
<tr>
<td>RU</td>
<td>5.64%</td>
<td>0.8016</td>
</tr>
<tr>
<td>Fusion</td>
<td>2.69%</td>
<td>0.3981</td>
</tr>
<tr>
<td><strong>Approach 1 (k=2)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZ-HU</td>
<td>4.07%</td>
<td>0.5661</td>
</tr>
<tr>
<td>CZ-RU</td>
<td>4.53%</td>
<td>0.6526</td>
</tr>
<tr>
<td>HU-RU</td>
<td>3.79%</td>
<td>0.5109</td>
</tr>
<tr>
<td>Fusion</td>
<td>2.27%</td>
<td>0.3393</td>
</tr>
<tr>
<td><strong>Approach 1 (k=3)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZ-HU-RU</td>
<td>4.34%</td>
<td>0.6500</td>
</tr>
<tr>
<td><strong>Approach 2 (k=2)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZ-HU</td>
<td>3.32%</td>
<td>0.4506</td>
</tr>
<tr>
<td>CZ-RU</td>
<td>3.58%</td>
<td>0.5276</td>
</tr>
<tr>
<td>HU-RU</td>
<td>2.75%</td>
<td>0.4140</td>
</tr>
<tr>
<td>Fusion</td>
<td>2.24%</td>
<td>0.3223</td>
</tr>
<tr>
<td><strong>Approach 2 (k=3)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZ-HU-RU</td>
<td>3.90%</td>
<td>0.5724</td>
</tr>
</tbody>
</table>

Mikel Penagarikano et al.  
Modeling of Cross-Decoder Phone Co-occurrences
Fused Systems Performance

<table>
<thead>
<tr>
<th>Fused Systems</th>
<th>EER</th>
<th>$C_{LLR}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>2.69%</td>
<td>0.3981</td>
</tr>
<tr>
<td>A1 (k=2)</td>
<td>2.27%</td>
<td>0.3393</td>
</tr>
<tr>
<td>A2 (k=2)</td>
<td>2.24%</td>
<td>0.3223</td>
</tr>
<tr>
<td>A1 (k=3)</td>
<td>4.34%</td>
<td>0.6500</td>
</tr>
<tr>
<td>A2 (k=3)</td>
<td>3.90%</td>
<td>0.5724</td>
</tr>
<tr>
<td>A1 (k=2) + A1 (k=3)</td>
<td>2.21%</td>
<td>0.3388</td>
</tr>
<tr>
<td>A2 (k=2) + A2 (k=3)</td>
<td>2.28%</td>
<td>0.3280</td>
</tr>
<tr>
<td>Baseline + A1 (k=2)</td>
<td>1.92%</td>
<td>0.3054</td>
</tr>
<tr>
<td>Baseline + A2 (k=2)</td>
<td>1.88%</td>
<td>0.3064</td>
</tr>
<tr>
<td>Baseline + A1 (k=3)</td>
<td>2.38%</td>
<td>0.3472</td>
</tr>
<tr>
<td>Baseline + A2 (k=3)</td>
<td>2.15%</td>
<td>0.3582</td>
</tr>
<tr>
<td>Baseline + A1 (k=2) + A1 (k=3)</td>
<td>2.02%</td>
<td>0.3056</td>
</tr>
<tr>
<td>Baseline + A2 (k=2) + A2 (k=3)</td>
<td>1.90%</td>
<td>0.3158</td>
</tr>
</tbody>
</table>
Introduction
Baseline SVM-based Phonotactic System
Cross-Decoder Phone Co-occurrences based System
Experimental Setup
Results
Summary

DET plots

LRE2007 eval (30s, closed)

Approach 1 (k=3)
Baseline
Approach 1 (k=2)
Baseline + Approach 1 (k=2)

LRE2007 eval (30s, closed)

Approach 2 (k=3)
Baseline
Approach 2 (k=2)
Baseline + Approach 2 (k=2)
$C_{\text{avg}}$ relative improvement per target language
Two approaches to the modeling of cross-decoder phone co-occurrences in SVM-based Phonotactic Language Recognition have been proposed and evaluated.

Both approaches outperformed the baseline system when using combinations of $k = 2$ decoders.

Co-occurrence information is more effectively extracted in 2-decoder configurations and recovered by means of fusion.

Under 3-decoder configuration, both approaches showed a poor performance compared to the baseline system. This may reveal robustness issues related to: the higher amount of transitional segments and the huge number of phone co-occurrence combinations.
Thank you!