Towards Spoken Term Discovery at Scale with Zero Resources

Aren Jansen   Kenneth Church   Hynek Hermansky

Human Language Technology Center of Excellence
Center for Language and Speech Processing

Interspeech 2010
Makuhari, Japan
Spoken Term Detection

Given a word or word sequence (term), find occurrences in continuous speech

Spoken Term Discovery

Find intervals of the speech signal that could potentially be terms of interest (i.e., spoken term detection without the terms)
What makes an interval of speech interesting?

The Problem

In $N$ frames of continuous speech, there are $\binom{N}{2}$ possible intervals.
What makes an interval of speech interesting?

The Problem

In $N$ frames of continuous speech, there are $\binom{N}{2}$ possible intervals

Cues from text processing:
- Length of interval
- Repetition of interval
- Localized frequency bursts (e.g. tf-idf)

Unique to speech processing:
- Fidelity of repetitions
- Lossy channel forces adjacent repetition
# The Zero Resource Case

## Zero Resources
No training data, no existing models, and no knowledge of linguistic structure for target language

## Zero Resource Term Discovery
Find repeated terms of possible interest in a speech stream without using any models or training data for the target language

---

**How well can you do?**

- Mexico
- Egypt

**Our zero resource system discovers:**

- 🎧聋
- 🎧聋
Past Approaches:

• Search for repeated trajectories in acoustic feature space
  MFCCs: [Park & Glass, 2005]; GMM Posteriors: [Zhang & Glass, 2010]
• Decode with unsupervised acoustic model (e.g. Herb Gish’s SOUs) and look for repeated phone-like unit sequences

Our Approach:

• Like [Zhang & Glass, 2010], but use native phonetic acoustic model (MLP-based posteriors) to interpret the foreign stream

Simplifying Assumptions

• Single speaker (for now)
• Fixed channel and/or robust front end
• Long terms of interest (∼ 1 second), as longer is easier
- $str = \text{‘text processing vs. speech processing’}$

- Dotplot is specified by the similarity matrix $M_{ij} = (str[i] == str[j])$

- Repeated substrings produce diagonal line segments
**From Text to Speech**

Instead of boolean similarity of characters, measure real-valued similarity between feature vectors.

---

**Fisher English: fe_03_04042**

(samples extracted automatically)
• Compute feature vector time series (e.g. spectrogram) 
\[ X = x_1x_2 \ldots x_N, \text{ where } x_i \in \mathbb{R}^d \]

• The acoustic dot plot \( M \) is defined as the Gram matrix

\[ M_{ij} = K(x_i, x_j) \equiv \langle x_i, x_j \rangle \]

• Repeated intervals of speech produce diagonal line segments
Step 1: Threshold and apply diagonal median filter of duration $\Delta t$
Pass 1: Efficient Line Segment Detection

Step 2: Apply the Hough transform (1D w/ fixed 45° angle)
Step 3: Use Hough transform peaks as starting points to perform line segment search
**Solution:** Apply second pass
Segmental DTW search for optimal nonlinear path

1. Run line segment search with shorter filter to find syllable-sized repeats

2. Start from the center of each repeat and find optimal curve using dynamic programming
Pass 2: Segmental-DTW [Park & Glass (2005)]

A More Fundamental Complication
Prosodic variability can easily break the line segment assumption

**Solution:** Apply second pass
Segmental DTW search for optimal nonlinear path

1. Run line segment search with shorter filter to find syllable-sized repeats

2. Start from the center of each repeat and find optimal curve using dynamic programming
A Complication
Computing dotplots require $O(N^2)$ memory space and CPU time

The Solution: Sparsity
- Sparse matrix storage and accompanying algorithms for:
  1. Diagonal median filtering
  2. Hough transform
  3. Line segment search
- Sparse acoustic features to reduce dotplot computation costs
The Posteriorgram Approximation

- Use sparse, MLP-based phone posteriorgrams as acoustic features to compute dotplots [Thomas, Ganapathy, Hermansky (2009)]
- Don’t waste time computing where the posteriorgram is close to zero (inverted file)

**Note:** Still $O(N^2)$, but the constants can be very small
Posteriorgram approximation introduces a **complication**:  

- Block structure from runs of a given speech sound  
- Long runs of silence and fill pauses induce massive blocks!  

**Solution:** Filter silence with speech activity detector and fill pauses according to posteriorgram stability  

Fisher English: fe_03_04042
Evaluation Setup

Switchboard (dev) & Fisher English (eval)

Conversational telephone speech corpora, each consisting of two party conversations (≈ 10 min files) on prompted topics

- Using transcriptions, define 1 target term per side such that
  1. Each target is repeated
  2. Each target has 3 uppercase letters (to ensure content)
     - e.g. Ed Sullivan Show, All Things Considered
  3. Each target is at least 1 second long
  4. Each target is the longest term (as text) in the file satisfying 1-3

- Apply term discovery algorithm to each conversation side

- Label all non-target matches as false alarms regardless of match quality

- Sort term discovery results by predicted duration
Results

- 75% recall after listening to 8 seconds (∼ 2%) of each file, without knowing the terms ahead of time.

Two-pass strategy can search 1 hour file in 6 machine-minutes! (300+ times faster than S-DTW alone [Zhang & Glass (2010)])
Results

- 75% recall after listening to 8 seconds (∼ 2%) of each file, without knowing the terms ahead of time

Two-pass strategy can search 1 hour file in 6 machine-minutes! (300+ times faster than S-DTW alone [Zhang & Glass (2010)])
Results

- 75% recall after listening to 8 seconds (~2%) of each file, without knowing the terms ahead of time

Two-pass strategy can search 1 hour file in 6 machine-minutes!
(300+ times faster than S-DTW alone [Zhang & Glass (2010)])
<table>
<thead>
<tr>
<th>File</th>
<th>1st Match</th>
<th>2nd Match</th>
<th>3rd Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>sw02102</td>
<td><strong>Ed Sullivan Show</strong></td>
<td>Ed Sullivan</td>
<td>the TV set</td>
</tr>
<tr>
<td></td>
<td><strong>Ed Sullivan Show</strong></td>
<td>Ed Sullivan</td>
<td>the TV set</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>1st Match</td>
<td>2nd Match</td>
<td>3rd Match</td>
</tr>
<tr>
<td>--------</td>
<td>---------------------------</td>
<td>----------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>sw02102</td>
<td>Ed Sullivan Show</td>
<td>Ed Sullivan</td>
<td>the TV set</td>
</tr>
<tr>
<td></td>
<td>Ed Sullivan Show</td>
<td>Ed Sullivan</td>
<td></td>
</tr>
<tr>
<td>sw02149</td>
<td>Big Brother Big Sister</td>
<td>thousand people without power</td>
<td>Peace Corps in Guatemala</td>
</tr>
<tr>
<td></td>
<td>Big Brother Big Sister</td>
<td>thousand people without power</td>
<td>Peace Corps in Guatemala</td>
</tr>
<tr>
<td>sw02499</td>
<td>like a Walt Disney film</td>
<td>critically acclaimed</td>
<td>The Incredible Journey</td>
</tr>
<tr>
<td></td>
<td>like a Walt Disney film</td>
<td>critically acclaimed</td>
<td>The Incredible Journey</td>
</tr>
<tr>
<td>sw04056</td>
<td>get twenty four days a year</td>
<td>vacation time</td>
<td>sick time</td>
</tr>
<tr>
<td></td>
<td>get twenty four days a year</td>
<td>vacation time</td>
<td>sick time</td>
</tr>
<tr>
<td>sw04326</td>
<td>it's a right to privacy</td>
<td>labs that are performing the</td>
<td>themselves and so</td>
</tr>
<tr>
<td></td>
<td>it was a right to privacy</td>
<td>lab that performs the</td>
<td>themselves i know</td>
</tr>
</tbody>
</table>

The predefined targets shown in **blue**

Notice: Most highly-ranked, non-target matches are reasonable
• Use **English** acoustic model to interpret foreign stream

• Acoustic model need not match input, but must preserve locality

• **Note:** Increased entropy in posterior distributions $\Rightarrow$ less computational savings from approximations

“the birth certificate and the marriage certificate”
(samples extracted automatically)
The Idea: Use discovered terms counts in standard bag-of-words document representation
Conclusions

1. Coarse-to-fine dotplot search can be used to efficiently detect terms of possible interest without predefining them.

2. High resource posteriorgrams can be used for speaker independent term discovery in zero resource settings.

3. Provides initial module to build zero resource spoken document retrieval and analysis systems.

References:


