Seamless Navigation in Audio Files

Christian Wellekens
Institut Eurécom, Sophia-Antipolis (F)
in collaboration with
Philippe Gelin
Lernout & Hauspie, Wemmel (B)
Perrine Delacourt
University College, Dublin (IRL)

Invited lecture at Speaker Odyssey 2001
Chania, Crete, June 21, 2001
OVERVIEW

• Why?
• How?
• Voice Activation Detection
• Keyword Spotting
• Segmentation in Speakers
Audio Data Indexing

- Facilitate and speed-up information retrieval in audio files (also contributes to audio visual indexing)
- Indexing key:
  - speaker identity
  - specific words
  - audio/speech versus silence/noise
Applications

- Indexing of databases
- TV news transcriptions system
- Speaker tracking
- ....
Technologies

- Voice activation detection
- Speech and music interval identification
- Keyword spotting
- Speaker identification
- Segmentation in speakers
Voice Activity Detection

• Why?
  – Speech enhancement (noise spectrum subtraction)
  – Speech recognition and speech database processing
  – Segmentation in speakers
  – Speaker recognition
  – Speech coding (no coding of non-speech: GSM)
Voice Activity Detection

• Principles
  – Parameter choice:
    * Entropy (Abdallah et al., 1997; Renevey, 2000)
    * Zero-crossing (Deller et al., 1993)
    * Statistical models (Wilpon & Rabiner, 1987)
  – Thresholding
Energy Based Detectors

- Energy and fixed threshold (off-line estimation of the statistical characteristics noise/speech)

**Problem**: minimum duration of speech intervals and pauses required
Energy Based Detectors

- Energy and adaptive threshold
  - Estimation of noise energy
    \[ E_{\text{noise}}(n) = \lambda_1 E_{\text{noise}}(n-1) + (1 - \lambda_1)E(n) \quad \text{in noise} \]
    \[ E_{\text{noise}}(n) = \lambda_2 E_{\text{noise}}(n-1) + (1 - \lambda_2)E(n) \quad \text{in speech} \]
  - Two thresholds (hysteresis)
    \[ T_n = E_{\text{noise}} + \delta_n \]
    \[ T_s = E_{\text{noise}} + \delta_s \]

Silence Detection (Delacourt 2000)

- Comparison of log-energy $\log E$ of a sample with the mean log-energy $\log E_{moy}$
- Hysteresis thresholding: 2 thresholds LOW and HIGH
Silence Detection

Labeling:
S = silence
P = speech

Duration:
VAD: conclusions

- Good detection of silence/noise (SNR high)
- Not suited for speaker turn detection:
  - speaker interventions should be separated by long silences
  - over-segmentation: segments are too short to be properly processed
Entropy Based VAD

• Parameters
  – short time spectrum
  – entropy

\[ H(S) = -\sum_{i=1}^{N} \frac{|S(f_i)|^2}{\sum_{j=1}^{N} |S(f_j)|^2} \log \frac{|S(f_i)|^2}{\sum_{j=1}^{N} |S(f_j)|^2} \]

• Assumption:
  – signal is more organized than noise
  – \( H(S) \) is maximum in white noise
Indexing Constraints on Keyword Detection

- Speaker independance
- Lexicon independance
- Searched word independance
- Speed
Keyword detection

- HMM models for KW and garbage problem:
  - limited number of keywords which should be known in advance
  - training of the garbage model
Keyword detection

- Adaptative garbage model
  - HMM with undefined emission probabilities
  - Emission probabilities selected at each frame among the top $N$ ($\leq 10$) but the best 5 emission probabilities for all states
  - Same probabilities for all states
  - Multiple states for duration modeling
Keyword detection

- Complete recognition of a full text and search in the recognized text

Problem:
- LVSIR required for spontaneous speech
- OOV problems
- Off-line phonetic lattice generation and search for phonetic transcription (Gelin 1997)
Keyword Detection
vs
Continuous ASR

- Spontaneous pronunciation
- Open lexicon
- Complex syntax
Used Scheme

\[ X = \{x_1, \ldots, x_N\} \]

OFF LINE

ON LINE

Time.

P(0.02)

PCL(0.2)

AXR(0.05)

EH(0.3)

ZH(0.2)

R(0.5)

AE(0.3)

Z(0.3)
Used methods

1) Frame labeling
   • models
   • lattice generation
   • lattice search

2) Markov models trained by MLE

3) Markov models trained by MAP
Frame Labeling

- 3 solutions:
  - Monogaussian:
    \[ p(x | \varphi) = \prod_{n=1}^{N} \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x_n - \mu_n)^2}{2\sigma^2}} \]
  - Multigaussian:
    \[ p(x|\varphi) = \sum_{d=1}^{D} P(G_{\varphi,d}|\varphi) P(x|G_{\varphi,d}) \]
  - Neural nets

\[ P(\varphi|x) = \frac{p(x|\varphi) P(\varphi)}{p(x)} \]
Lattice generation: smoothing

- LP filtering of probability trajectories as a function of the average duration of phonemes $d(\varphi)$:

$$P_{\text{int}}(t, \varphi) = \frac{1}{d_{\varphi}} \sum_{n=t-d_{\varphi}/2}^{t+d_{\varphi}/2} P(\varphi | x_n)$$
Lattice generation : Detection

\[ P(\phi | x_t) \]

\[ P(\phi | X_b^e) = \frac{\prod_{t=b}^e P(\phi | x_t)}{P(\phi)^{e-b}} \]

thresholds

\[ h_1 \]

\[ h_2 \]
Lattice Generation: transitions

Frames

0 2 4 6 8 10 20 30
Search : Algorithm

• Initialization
  • search of the first phoneme
  
  \[ H = h_{l_1}(\tilde{\varphi}_1, P, b_1, e_1) \quad P(H) = P(h_{l_1}) \]

• Recurrence
  • search of the next phonemes
  
  \[ H = \{H, h_{l_{n+1}}\} \quad P(H) = P(H) \ast P(h_{l_{n+1}}) \]

such that : \( b_n \leq b_{n+1} \leq e_n \) et \( e_n \leq e_{n+1} \)

• or : \( e_n \leq b_{n+1} \leq e_n + Cst \)
Used Methods

1) Frame Labeling
2) Markov models trained by MLE
   • Models
   • Lattice Generation
   • Search
   • Results
3) Markov models trained by MAP.
\[ P(\varphi_j | \varphi_i) = \sum_v N_{\text{transition}}(\varphi_i, \varphi_v) \]
Lattice Generation

\[ P(\varphi | X_{t_b}^e) = \frac{P(\varphi)}{C^{e-b+1}} \prod_{t=b}^{e} P(x_t | q_{\varphi_v}^t) P(q_{\varphi_v}^{t+1} | q_{\varphi_v}^t) \]

\[ P(\varphi | X_{t_b}^t) = P(\varphi | X_{t_b}^t)^{\frac{t-t_b+1}{t-t_b+1}} P(\varphi)^{1-\frac{t-t_b+1}{t-t_b+1}} \]
Lattice Search

• Initialization
  • Search for the first phonemes
    \[ H = h_{l_1}(\tilde{\phi}_1, P, b_1, e_1) \]
    \[ P(H) = P(h_{l_1}) \]

• Recurrence
  • Search for the next phonemes
    \[ H = \left\{ H, h_{l_{n+1}}(\tilde{\phi}_{n+1}, P, b_{n+1}, e_{n+1}) \right\} \]
    \[ P(H) = P(H) \ast P(h_{l_{n+1}}) \]
  
    such that \( b_{n+1} = e_n \)
1) Frame labeling
2) Markov models trained by MLE
3) Markov models trained by MAP

- Model
- Generation
- Search
**REMAP**

- **Principle**
  - Maximization of a posteriori probabilities
    \[
    \arg \max_i P(M_i | X, \lambda)
    \]
  - Use of a discriminant criterion
    \[
    \sum_{k=1}^{K} P(q_k^n | q_i^{n-1}, X_{n-c}^{n+d}, \lambda) = 1 \Rightarrow \sum_{i=1}^{I} P(M_i | X_i, \lambda) = 1 \quad \forall i, n
    \]
REMAP: Use

\[ P\left(q_k^t | q_j^{t-1}, X_{t-c}^{t+d}\right) \]

Neural Network

\[ P\left(X_{t+d}^T | q_i^t, X_1^t, M\right) \]

\[ \alpha_i^t = P\left(q_i^t, X_1^t | M\right) \]
\[ = \sum_{k=1}^{K} \alpha_k^{t-1} P\left(q_i^t | q_k^{t-1}, X_{t-c-1}^{t+d-1}, M\right) c_k^t \]

\[ \gamma_i^t = P\left(X_{t+1}^T | q_i^t, X_1^t, M\right) \]
\[ = c_i^{t+1} \sum_{k=1}^{K} P\left(q_k^{t+1} | q_i^t, X_{t-c}^{t+d}, M\right) \gamma_k^{t+1} \]

\[ P\left(M | X\right) = \frac{P(M) \sum_{k=1}^{K} \alpha_k^t \gamma_k^t}{P(X)} \]
REMAP : Training

\[ P(q_i \mid q_{i-1}, X, M, \lambda) \]

Update

\[ P(q_i \mid q_{i-1}, X, M, \lambda') \]

Neural Network \( \lambda \)

\[ q_i \]

\[ X_{t-1+d} \]

Data base

\[ P(q_{i-1} \mid X, M, \lambda) \]
REMAP: Lattice Generation

- Search for the best path

\[ v_i^t = \max_k v_k^{t-1} P(q_i^t | q_k^{t-1}, X^{t+d}_{t-c}, M)c_k^t \]

- Probability associated with a hypothesis

\[ P((\varphi_k)_b | X, M) = P(q_b^b | q_{k-1}^b, X^{b+d}_{b-c}, M) \prod_{t=b+1}^{e} P(q_k^t | q_k^{t-1}, X^{t+d}_{t-c}, M) \]
Lattice search

- Initialization
  - search for the first phonemes
  \[ H = h_t(\tilde{\varphi}_1, P, b_1, e_1) \quad P(H) = P(h_t) \]

- Recurrence
  - search for the next phonemes
  \[ H = \{ H, h_{t+1}(\tilde{\varphi}_{n+1}, P, b_{n+1}, e_{n+1}) \} \quad P(H) = P(H) \cdot P(h_{t+1}) \]
  - such that \( b_{n+1} = e_n \)
Segmentation in speakers (Delacourt 2000)

Hypotheses:
- no available information on speakers or on language (neither speaker model, nor speech model, ...)
- unknown number of speakers
- only speech data: no music, no advertising...
- people do not speak simultaneously
- no real-time constraints
Speaker-based segmentation

↑ to extract homogeneous segments containing the longest possible utterances produced by a single speaker

Several types of speaker-based segmentations:

– segmentation based on silence detection
– segmentation based on speaker turn detection
– segmentation based on speaker models
Speaker turn detection: 1st pass

- First pass: distance-based segmentation

\[ \text{\textbf{\Large \textbf{\text少}}} \] computation of a distance between two portions of parameterized signal:

- low value \( \Rightarrow \) 1 speaker
- high value \( \Rightarrow \) 2 speakers

Sequence of acoustic vectors:

```
Win 1 (time t) | Win 2 (time t)
Win 1 (time t+1) | Win 2 (time t+1)
Win 1 (time t+2) | Win 2 (time t+2)
```

\ldots
Speaker turn detection: distance used

- Hypothesis test:
  - $H_0$: both segments are related to the same speaker
  - $H_1$: segments are generated by different speakers

- Generalized likelihood ratio [GISH91]:

$$\text{GLR} = \frac{L_0}{L_1} = \frac{L(\zeta, N(\mu, \Sigma))}{L(\chi, N(\mu_1, \Sigma_1)) L(\psi, N(\mu_2, \Sigma_2))}$$

$$R = -\log \text{GLR}$$
Detection of speaker turns

Detected speaker turns

Magnitude difference higher than a threshold

2 consecutive maxima should be separated by a minimal time interval

\[ \alpha \sigma \]
Speaker turn detection: 2\textsuperscript{nd} pass

- Second pass: refinement of the segmentation by reducing the number of false alarms.

**Diagram:**
- Actual segments
- Resulting segments
- Missed detection
- False alarm
Speaker turn detection: use of the BIC

- Bayesian Information Criterion (BIC): likelihood criterion penalized by the model complexity [Chen 98]

\[
\text{BIC}(M) = \log L(\chi, M) - \lambda \frac{m}{2} \log N
\]

$L(\chi, M)$: likelihood of the sequence of acoustic vectors $\chi$ according to the model $M$

$N$: number of acoustic vectors in the sequence $\chi$

$m$: number of parameters to estimate

$\lambda$: penalty factor
Speaker turn detection: use of the BIC

- BIC difference: \( \Delta \text{BIC} = -R + \lambda P \) with \( P = \frac{1}{2} (d + \frac{1}{2} d(d+1)) \log N \)

- Maximum likelihood ratio:

\[
R = \frac{L_0}{L_1} = \frac{N}{2} \log |\Sigma| - \frac{N_1}{2} \log |\Sigma_1| - \frac{N_2}{2} \log |\Sigma_2|
\]

\( \Delta \text{BIC} < 0 \) → a speaker turn is detected
Speaker turn detection: recap

- First pass: distance-based segmentation
- Second pass: BIC refinement

1st case: \( \Delta \text{BIC} > 0 \) \( \checkmark \) the speaker turn is discarded

2nd case: \( \Delta \text{BIC} < 0 \) \( \checkmark \) the speaker turn is validated
Speaker turn detection: assessment

False Alarm Rate: \[ \text{FAR} = 100 \times \frac{\text{# of FA}}{\text{# of actual turns} + \text{# of FA}} \%
\]

Missed Detection Rate: \[ \text{MDR} = 100 \times \frac{\text{# of MD}}{\text{# of actual turns}} \%
\]

actual segments

resulting segments

missed detection (MD)

false alarm (FA)
Results: Data and Parameterization

• Data:
  – 2 synthetic conversations in English language (TIMIT, clean speech, short segments, 60 speaker turns)
  – 2 synthetic conversations in French language (CNET, clean speech, short segments, 45 speaker turns)
  – 3 French language TV news (INA, segments of any length, 85 speaker turns)
  – 3 phone conversations (SWITCHBOARD, segments of any length, 120 speaker turns)
  – 3 French language TV news (JT, collected in our lab, segments of any length, 830 speaker turns)

• Parameterization:
  – 12 Mel-cepstral coefficients
### Segmentation: results

1rst pass provides better results for short segments

<table>
<thead>
<tr>
<th></th>
<th>distance-BIC : distance</th>
<th>distance-BIC : BIC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FAR</td>
<td>MDR</td>
</tr>
<tr>
<td>TIMIT</td>
<td>40.3 %</td>
<td>14.3 %</td>
</tr>
<tr>
<td>CNET</td>
<td>18.2 %</td>
<td>16.7 %</td>
</tr>
<tr>
<td>INA</td>
<td>37.4 %</td>
<td>9.0 %</td>
</tr>
<tr>
<td>SWITCHBOARD</td>
<td>39.0 %</td>
<td>29.1 %</td>
</tr>
<tr>
<td>JT</td>
<td>59.0 %</td>
<td>8.9 %</td>
</tr>
</tbody>
</table>

Errors happen during interviews

Fast exchanges

Significant reduction of false alarms
Segmentation : conclusions

• Parameters depend on the actual length of speaker segments

• Not easy to compare with other segmentation methods: Word Error Rate in most evaluations

• In most applications, over segmentations is preferable
Speaker-based clustering

- To group segments by speakers (cluster = segment resulting from the previous step)
  - merging criterion ?
  - stopping criterion ?
Hierarchical clustering (top-down)

- The two nearest clusters, according to the merging criterion, are merged and the process is repeated until the stopping criterion is reached.
S. Chen’s method

- Merging criterion: likelihood rate
- Stopping criterion: Bayesian Information Criterion

\[ R = -\log \frac{L(\zeta, N(\mu, \Sigma))}{L(\chi, N(\mu_1, \Sigma_1))L(\psi, N(\mu_2, \Sigma_2))} \]

\[ \Delta \text{BIC}(k, l) = -R(k, l) + \lambda P \]

- \( \Delta \text{BIC}(k, l) < 0 \) \( \Rightarrow \) Stop of clustering
- \( \Delta \text{BIC}(k, l) \geq 0 \) \( \Rightarrow \) Segments \( k \) et \( l \) are clustered
Clustering: assessment methods

• Number of resulting clusters:
  
  \[ \text{# of clusters} = \text{# of speakers} \]

• Cluster purity:

  \[ p = \frac{\text{# of segments of the majority speaker}}{\text{# of speaker segments in cluster}} \]
Clustering : reference data

• Data :
  - 10 synthetic english conversations (TIMIT, clean speech, short segments, 52 minutes)
  - 10 synthetic french conversations (CNET, clean speech, short segments, 34 minutes)
  - 2 synthetic french dialogs (CNET, long segments, 26 minutes) + same 2 without silences
  - 2 synthetic french dialogs (CNET, long segments, 60 minutes) + same 2 without silences

• Parameterization :
  - 16 Mel-cepstral coefficients
Clustering: reference segments

- Actual # of speakers
- Found # of speakers
- Purity (in %)
- Duration (in sec.)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Actual # of speakers</th>
<th>Found # of speakers</th>
<th>Purity (%)</th>
<th>Duration (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNET</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIMIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONV</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Clustering : actual data

• Data :
  – 10 synthetic english conversations (TIMIT, 52 mn)
  – 10 synthetic french conversations (CNET, 34 mn)
  – 2 synthetic french dialogs (CNET, 26 mn)
  – 2 synthetic french dialogs (CNET, 60 mn)
  – 3 french TV news (laboratory, 126 mn)
  – 49 phone calls in english (SWITCHBOARD, 5 to 10 mn per conversation)

• Parameterization :
  – 16 Mel-cepstral coefficients
Comparison reference/actual segments

reference = without segmentation
sil+dist = segmentation with pre-processing
dist = segmentation

Actual # of speakers
Found # (reference)
Found # (sil+dist)
Found # (dist)
Purity (reference)
Purity (sil+dist)
Purity (dist)
Duration (reference)
Duration (sil+dist)
Duration (dist)
Clustering: actual conversations

- **Type I**: one speaker \(\forall\) recording conditions
- **Type II**: one speaker/recording condition

- **sil+dist**: segmentation with pre-processing
- **dist**: segmentation

<table>
<thead>
<tr>
<th></th>
<th>JT (Type I)</th>
<th>JT (Type II)</th>
<th>SWB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual # of speakers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Found # (sil+dist)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Found # (dist)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purity (sil+dist)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purity (dist)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duration (sil+dist)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duration (dist)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>