Speaker Diarization Tutorial

SCALE workshop, 01/13/2010
Christian Müller, DFKI
Outline

- Introduction to the agglomerative clustering approach as an example of a typical state-of-the-art diarization system.
- Detection of overlap speech and run-time optimization as examples of current issues.
- Incorporation of long-term (high level) features for improving accuracy.
- Differences between offline and online speaker diarization
- Downstream application examples
What is Speaker Diarization?

- The goal of speaker diarization is to automatically segment an audio recording into speaker homogeneous regions.
- The identity of each speaker is not known and even the number of speakers is unknown.
- The system is supposed to anonymously label each speaker in the recording and answer the question: ‘Who spoke when?’
Speaker Diarization Concept

Audiotrack:

Segmentation:

Clustering:
A Typical Situation: Meetings

- Participants are not known (no speaker model)
- Speaker may stand up and move around
- There may or may not be close-talk microphones
- Room characteristics will be crucial (e.g. reverberation)
- Spontaneous speech is likely (disfluencies, overlap, short turns)
A Typical System

- A typical system is based on Hidden Markov Models (HMM) with GMMs as probability density functions.
- The HMM states drawn horizontally all states share a single GMM
- In an ideal situation, each GMM is trained on all the speech of one unique speaker
- The speaker segmentation, the final system result, is found by performing a Viterbi alignment of all audio that contains speech
- All audio that is processed by the same string of states during this alignment is grouped together as speech from one speaker

Initial Training

- Obviously, the initial state of the system will not automatically be the preferred situation (where each speaker is represented by exactly one string of states).
- Instead, too many HMM states are created.
- The number of states is then iteratively decreased and the GMMs are slowly trained on speech from a single speaker until the correct number of GMMs is reached.
- On the following slides, the respective algorithm is explained.
The system will be initialized with a large number of models (strings of HMM states).

The number of models should be significantly higher than the assumed maximum number of speakers in the audio file (e.g. 16).

The available speech data is cut up into small pieces and these pieces are randomly divided in a number of bins.

Each bin is used to train one of the GMMs (e.g. 5 mixtures).

Although the GMMs are trained using multiple speakers, in general one speaker will fit the GMM a little bit better than the other speakers.

Therefore, when a Viterbi alignment is performed, this GMM will be assigned more speech from this speaker.
Iteration Algorithm II

- The data will be re-aligned a number of times and after each iteration, the GMMs are re-trained.
- Each iteration the model will fit the dominant speaker better than before.
- The result of this step is a group of models that are all trained with as much data as possible of one dominant speaker and as little data as possible of the other speakers.
- In the remaining steps, the models that are trained on the same dominant speaker will need to be merged.
Merging

- In the third step it is determined which two models are most likely trained on the same speaker.
- This is done by calculating the local Bayesian Information Criterion (BIC) score for each combination of two models.
- For this BIC comparison, a new model is trained containing the sum of the number of Gaussians of the two original models a and b.
- This merged model $\theta$ is trained on the training data of the original two models.
- If the two models are trained on data of one single speaker, this merged model must be able to replace the two models without decreasing system performance.
- The following formula is used to calculate this change in system performance. $D_a$ is the data used to train model a, $D_b$ to train b and $D$ to train model $\theta$.

$$BIC(\theta_a, \theta_b) = \log P(D|\theta) - \log P(D_a|\theta_a) - \log P(D_b|\theta_b)$$
Merging II

- If the BIC score is positive, a and b are considered to be trained on data of the same speaker.
- The higher the BIC score, the more the two models were similar.
- Therefore, the cluster pair with the highest BIC score will be chosen as the candidate for merging.
- The decision to merge the candidate cluster pair is positive if the BIC score is bigger than zero and negative if the BIC score is negative.
- The two merge candidates are replaced in the HMM by the merged model.
- The data is re-aligned over the models and the models are re-trained with the new data.
- After this, a new merging iteration is started.
- If no BIC score greater 0 is found, Viterbi alignment will be performed and the algorithm is finished.
Schematic Overview

Speech Activity Detection

Create 16 models

Pick best models to merge

Merge? -> Merge models

Create models

Re-align data

Train models (x iterations)

N times

merge and train model

Re-align data

Train all models (x iterations)

N times

Done!
Parameters

- The speaker diarization system is designed to have no system parameters that need to be tuned on external data, making it robust for changes in audio conditions or application domain.

- Because the GMM models are trained on the data under evaluation, no models created on a training set are needed.

- Unfortunately, a number of system parameters still exist, although these parameters do not seem sensitive for changes in audio conditions.
  - Number of initial clusters (16)
  - Number of Gaussians (5)
  - Minimum segment duration (2.5 s)
  - State transition penalties (0)
The “Blame Game” or What Makes an Audio Recording Hard to Diarize Correctly?

<table>
<thead>
<tr>
<th>Test description</th>
<th>DER (%)</th>
<th>Relative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overlapping speech</td>
<td>2.60</td>
<td>22.11</td>
</tr>
<tr>
<td>Speech Activity Detection</td>
<td>2.70</td>
<td>22.96</td>
</tr>
<tr>
<td>Modeling/alignment</td>
<td>1.57</td>
<td>13.35</td>
</tr>
<tr>
<td>Merging algorithm</td>
<td>0.49</td>
<td>4.17</td>
</tr>
<tr>
<td>Non-perfect initial clusters</td>
<td>1.82</td>
<td>15.48</td>
</tr>
<tr>
<td>Combining wrong models</td>
<td>1.22</td>
<td>10.37</td>
</tr>
<tr>
<td>Stop clustering too early/late</td>
<td>1.36</td>
<td>11.56</td>
</tr>
<tr>
<td>System DER (sum of components)</td>
<td>11.76</td>
<td>100.00</td>
</tr>
</tbody>
</table>

Data: twelve conference meetings from previous NIST Rich Transcription benchmark evaluations
Issues Part 1: Overlap Speech

- The presence of overlapped, or co-channel, speech in meetings is a common occurrence and a natural consequence of the spontaneous multiparty conversations.

- Current state-of-the-art systems assign speech segments to only one speaker, thus incurring missed speech errors in regions where more than one speaker is active.

- As overlapped speech is now a major obstacle in improving the performance of speaker diarization systems, efforts in overlap detection will be of increasing interest and importance.

- A typical overlap detector is an HMM-based segmenter that operates using features tailored for the task.

- The post-processing procedure is a speaker assignment method for the identified overlap segments based on speaker posterior probabilities produced by the diarization system.

"Overlapped Speech Detection for Improved Speaker Diarization in Multiparty Meetings" K.A. Boakye, B. Trueba-Hornero, O. Vinyals, and G. Friedland
April 2008
As with any detection scheme, the overlap system is susceptible to errors of two types: false alarms and misses.

These errors impact the diarization system quite differently

- false alarms are carrying through to increase the diarization false alarm error
- misses are having no effect on the baseline diarization error.
These errors impact the diarization system quite differently:
- false alarms are carrying through to increase the diarization false alarm error.
- misses are having no effect on the baseline diarization error.

Because of this difference, the overlap detector is optimized for low false alarms, which corresponds to a high precision (and possibly low recall) operating point.
Example: HMM-based Overlap Segmenter

- The segmenter consists of three classes
  - non-speech
  - Speech
  - overlapped speech
- Each class is represented with a three-state model.
- State emission probabilities are modeled using a multivariate Gaussian Mixture Model (GMM) with (in this case) 32 components and (in this case) diagonal covariance matrices.
- Classes are identified in the training data using ASR forced-alignment times generated from ground-truth transcriptions of the audio.
- Test audio signals are segmented into regions labeled as one of the three classes using a single Viterbi decoding pass.
Overlap Detection Features

- A key consideration in the overlap detection system is the selection of features used in the HMM-based segmenter.
- Baseline MFCCs
- The short-time root-mean-squared (RMS) energy
  - The energy content of a speech segment will likely be affected by the presence of additional speakers
  - Signal waveforms are normalized based on overall RMS channel energy estimates.
- LPC residual energy
  - Linear predictive coding (LPC) coefficients encode the formants of a speaker
  - Residual signal represents the portion that cannot be attributed to this formant model—typically the excitation source.
  - In the case of more than one speaker, a fixed-order LPC representation will not be able to model the spectrum (shaped by formants of multiple speakers) well.
  - This potentially leads to more energy content in the residual signal
Overlap Detection Features II

- Diarization posterior entropy
  - Using frame-level speaker likelihoods from the diarization system, the posterior probability for each speaker on every frame and subsequently a frame-level entropy from these posteriors is computed.
  - In single-speaker regions one model is expected to have the highest probability and the remainder to have significantly lower values.
  - In overlap segments there should be lower, more evenly distributed probabilities among the overlapping speakers and, as a result, the entropy should be higher.
Diaperization Segment Post-Processing

- Having identified regions of overlapped speech, this information can then be used to modify segment and label information output by the diarization system.
- In an overlapped segment, the frame-level speaker posteriors are summed over the frames of the segment to obtain a single “score” for each speaker.
- Typically the diarization system will have assigned the segment to the speaker with the highest score, in which case the speaker with the second highest score is chosen as the other speaker.
- In the event that the system has chosen another speaker, then this highest scoring speaker is selected as the additional speaker.
- this procedure limits the number of possible overlapping speakers to two (typically comprises 80% or more of the instances of overlapped speech).
Schematic Overview

DIARIZATION ENGINE

OVERLAP DETECTOR

POST PROCESSING

SCALE speaker diarization tutorial, 01/13/2010, Christian Müller
## Empirical Results

<table>
<thead>
<tr>
<th>System</th>
<th>Prec.</th>
<th>Rec.</th>
<th>F-score</th>
<th>DER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline Diarization</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>32.28</td>
</tr>
<tr>
<td>MFCC+Δ</td>
<td>0.7</td>
<td>0.27</td>
<td>0.39</td>
<td>30.84</td>
</tr>
<tr>
<td>MFCC+Eg+LPC+DPEΔ</td>
<td>0.72</td>
<td>0.25</td>
<td>0.37</td>
<td>30.46</td>
</tr>
<tr>
<td>MFCC+DPE+Δ</td>
<td>0.73</td>
<td>0.32</td>
<td>0.45</td>
<td>30.13</td>
</tr>
<tr>
<td>MFCC+Eg+DPE+Δ</td>
<td>0.76</td>
<td>0.34</td>
<td>0.47</td>
<td>29.90</td>
</tr>
</tbody>
</table>

AMI development data, near-field audio

<table>
<thead>
<tr>
<th>System</th>
<th>Prec.</th>
<th>Rec.</th>
<th>F-score</th>
<th>DER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline Diarization</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>38.11</td>
</tr>
<tr>
<td>MFCC+Δ</td>
<td>0.54</td>
<td>0.15</td>
<td>0.24</td>
<td>38.09</td>
</tr>
<tr>
<td>MFCC+Eg+LPC+DPE+Δ</td>
<td>0.61</td>
<td>0.33</td>
<td>0.42</td>
<td>37.26</td>
</tr>
<tr>
<td>MFCC+DPE+Δ</td>
<td>0.64</td>
<td>0.31</td>
<td>0.42</td>
<td>36.83</td>
</tr>
<tr>
<td>MFCC+Eg+DPE+Δ</td>
<td>0.66</td>
<td>0.26</td>
<td>0.37</td>
<td>36.75</td>
</tr>
</tbody>
</table>

AMI development data, far-field audio
The most successful SD approaches, based on agglomerative clustering, exhibit an inherent computational complexity which makes real-time processing, especially in combination with further processing steps, almost impossible.

A major runtime bottleneck of state-of-the-art systems (as described) is the computation of the natural logarithm.

This comes at no surprise because it relies heavily on the computation of logarithms because it uses log-likelihoods as a basic similarity measure.

Profiling result: computing the log-likelihood takes about 80% of the total runtime.

Many implementations of the logarithm function are either too slow, too inaccurate, or require special hardware.

ICSILog Algorithm

- An IEEE 754 floating point number is decomposed into mantissa and exponent.
- The mantissa is quantized and used as a pointer into a lookup table that fits into CPU cache.
- The result of the look up can be easily composed with the downscaled exponent using one addition.
## Speed-Up Factor

<table>
<thead>
<tr>
<th></th>
<th>Standard Log</th>
<th>Fast Log</th>
<th>ICSI Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time needed</td>
<td>100%</td>
<td>49%</td>
<td>45%</td>
</tr>
<tr>
<td>Diarization Error Rate</td>
<td>11.74</td>
<td>12.14</td>
<td>11.74</td>
</tr>
</tbody>
</table>

Speed-up factor of the ICSILog relative to the GCC standard log implementation.

- Fast speaker diarization approach: fast-match component to largely reduce the hypothesis space of the BIC-based model selection.
- The basic idea is basic idea of fast-match is using a computationally cheap method to reduce the hypothesis space of the more expensive and accurate search.
- Fast-match is essentially a search space tailoring technique.

<table>
<thead>
<tr>
<th>Component</th>
<th>Run-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find Best Merge Pair and Merge</td>
<td>62 %</td>
</tr>
<tr>
<td>Model Re-training/Re-alignment</td>
<td>28 %</td>
</tr>
<tr>
<td>Other</td>
<td>10 %</td>
</tr>
<tr>
<td>Total</td>
<td>100 %</td>
</tr>
</tbody>
</table>

Run-time distribution of the ICSI speaker diarization system.

“A Fast-Match Approach for Robust, Faster than Real-Time Speaker Diarization”
The first strategy uses the pitch-correlogram, as a type of prosodic feature, to capture speaker variances by looking at the pitch patterns.

In the second strategy we use KL-divergence, as a measurement of the difference between two probabilistic distributions.

Based on these two strategies, light-weight scoring schemes are developed to measure how likely two clusters are to be merged before applying the more expensive model selection via BIC.

The technique can reduce the hypothesis space by 86% and speed up the system by 41%, without affecting the speaker diarization error rate.
Schematic Overview

Cluster 1 \rightarrow \text{Cluster 2} \rightarrow \text{Cluster 3} \rightarrow \text{Cluster 4} \rightarrow \text{Cluster 5} \rightarrow \ldots \rightarrow \text{Cluster n}

\begin{align*}
\text{Fast Match} \\
\text{Bayesian Information Criterion}
\end{align*}

\begin{align*}
\text{New} \\
m \text{Comparisons, } m \ll \binom{n}{2}
\end{align*}

\binom{n}{2} \text{ Comparisons}

\text{Merging Decision } \quad \text{e.g. Cluster 1 \rightarrow Cluster 5}
The pitch-correlogram is used to capture pitch dynamics by looking at the statistics of pitch patterns at frame level distance.

Specifically, a pitch-correlogram \((H)\) is the joint distribution of quantized pitch bands explored at certain frame level distances \((k)\). When \(k\) is set to 1, the pitch-correlogram is a 2-dimensional table \(H = [h_{ij}]\), which basically collects the bigram statistics of the quantized pitch of neighboring frames.

\[
C_{ij} = \frac{\text{Number of times pitch band } P_i \text{ is succeeded by } P_j}{\text{Total number of voiced frames} - 1},
\]

110 bands, 50-500Hz
KL-Divergence

- It measures how likely it is that two models are to be merged by asking the question, “how different are these two distributions”
- The KL-divergence between two GMMs is approximated using the highly accurate and efficient unscented transform.

$$KL(f(x) || g(x)) = \int f(x) \log \frac{f(x)}{g(x)} dx$$
## Speed-Up

<table>
<thead>
<tr>
<th></th>
<th>DER</th>
<th>#BIC</th>
<th>$T_{SC}$ (s)</th>
<th>$xR_{T1}$</th>
<th>$xR_{T2}$</th>
<th>SU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>11.74%</td>
<td>4951</td>
<td>11173</td>
<td>1.60</td>
<td>1.45</td>
<td>NA</td>
</tr>
<tr>
<td>$PC_r=0.1$</td>
<td>11.49%</td>
<td>2035</td>
<td>8727</td>
<td>1.25</td>
<td>1.15</td>
<td>22%</td>
</tr>
<tr>
<td>$PC_{Top5}$</td>
<td>12.53%</td>
<td>697</td>
<td>6667</td>
<td>0.96</td>
<td>0.90</td>
<td>40%</td>
</tr>
<tr>
<td>$KL_r=0.1$</td>
<td>12.52%</td>
<td>2060</td>
<td>8347</td>
<td>1.20</td>
<td>1.10</td>
<td>25%</td>
</tr>
<tr>
<td>$KL_{Top5}$</td>
<td>11.58%</td>
<td>715</td>
<td>6570</td>
<td>0.88</td>
<td>0.94</td>
<td>41%</td>
</tr>
</tbody>
</table>

Results of pitch-correlogram and KL-divergence fast-match (SU is the speed-up of $T_{SC}$ over the baseline)
Speed-Up Summary

ICSILog + Code optimization +
+ FastMatch KL div, DER 11.58

Real Time Barrier

ICSILog + Code optimization 1, DER 11.74
ICSILog + Code optimization 2, DER 11.74
ICSILog, DER 11.74

Baseline, DER 11.74

Relative Speed Up

Time

SCALE speaker diarization tutorial, 01/13/2010, Christian Müller
Advances: Long-Term Features

- In the related field of speaker recognition, task-specific features have been successfully applied in combination with MFCCs.
- These features are often obtained on portions of speech longer than one frame and are therefore referred to as long-term features.
- Short-term cepstral features are generally referred to as *low-level features* reflecting the voice parameters of the speaker as opposed to *higher-level features* that capture phonetic, prosodic, and lexical information.

Prosodic and Other Long-Term Features for Speaker Diarization
G. Friedland, O. Vinyals, Y. Huang, and C. Müller
IEEE Transactions on Audio, Speech, and Language Processing, Vol. 17, No. 5, pp. 985-993
High-level features (learned characteristics)

Low-level features (physical characteristics)

Hierarchical Feature Model

<=> how shall I say this <c> <s> yeah I know...
Advances: Long-Term Features

- In speaker recognition it was shown that systems using a combination of cepstral and higher-level features outperform standard systems, especially when the amount of available training data was increased.

- This confirms the assumption that short-term cepstral systems generally perform well because they reflect information about the speaker’s physiology and do not rely on the phonetic content (which makes them inherently text-independent).

- However, long range information that also resides in the signal is only exploited in the combined systems.

- Also, higher-level features also have the potential of increased robustness to channel variation, since lexical usage or temporal patterns do not change with the change of acoustic conditions.
Feature Selection

- The list of initial candidate features can be assigned to five different categories: pitch, energy, formants, harmonics-to-noise ratio, and long-term average spectrum.
Feature Selection

- In total, the list of initial candidate features had 52 elements.
- To obtain a smaller set of features, their general speaker discriminability was estimated in a pre-experiment using the TIMIT database.
- The relative speaker discriminability was estimated on the basis of the ratio of the within-speaker variability ($w$) and the between-speaker variability ($b$), where $\max(b/w)$ indicates the best feature in the test.
- The median and average fundamental frequency were the best features, followed by high formants (F4, F5). Also, the mean harmonics-to-noise ratio and the variance of the long-term average spectrum achieved a high rank.
## Empirical Results

<table>
<thead>
<tr>
<th></th>
<th>ICSI SDM System at Eval07</th>
<th>ICSI Devset 07</th>
<th>Eval07</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech/Non Speech Error</td>
<td></td>
<td>6.4%</td>
<td>6.8%</td>
</tr>
<tr>
<td>Speaker Error</td>
<td></td>
<td>11.3%</td>
<td>14.9%</td>
</tr>
<tr>
<td><strong>Diarization Error Rate</strong></td>
<td></td>
<td>17.57%</td>
<td>21.24%</td>
</tr>
<tr>
<td>Current System (using prosodic features)</td>
<td>ICSI Devset 07</td>
<td>Eval07</td>
<td></td>
</tr>
<tr>
<td>Speech/Non Speech Error</td>
<td></td>
<td>6.2%</td>
<td>6.8%</td>
</tr>
<tr>
<td>Speaker Error</td>
<td></td>
<td>7%</td>
<td>9.5%</td>
</tr>
<tr>
<td><strong>Diarization Error Rate</strong></td>
<td></td>
<td>13.29%</td>
<td>16.28%</td>
</tr>
</tbody>
</table>

- top-10 long term features as additional feature stream improves DER by 35% relative (5% absolute).
Empirical Results II

- It was also to be seen that the top-ten prosodic features contribute especially in the last stages of the agglomerative clustering approach.
- the effect of the prosodic features on the first iterations is unnoticed by the algorithm: the MFCCs alone are able to refine the segments and merge clusters that belong to the same speaker.
- As the clusters are merged, the average length increases and thus the long-term dependencies that the prosodic features extract are more robust.
- Moreover, in the last stages of the algorithm the clusters are more pure (each cluster contains speech from only one person), and, as a consequence, the discriminative power that the prosodic features have is amplified by the fact that the clusters represent speech from mostly one person.
Online “Live” Speaker Diarization

- The output of conventional speaker diarization is limited to labeling speaker regions with numbers or letters, but not with real names.
- Also, they require the processing of entire files and thus do not work online.
- The goal of online speaker diarization is to answer the question “Who is speaking now?”
- For the system to perform live identification, the question has to be answered on small chunks of the recorded audio data, and the decisions must not take longer than real-time.

"Live Speaker Identification in Conversations"
Example System
In training mode, the user is asked to speak for 60 seconds. For the recognition to work properly, about 45 seconds of pure speech (i.e. no pauses) are needed.

The speech segments are then concatenated and used to train a Gaussian Mixture Model (GMM).
Example System

- In recognition mode, the system constantly records audio and processes it.
- For every frame, the likelihood for each set of features is computed against each set of Gaussian Mixtures obtained in the training step, i.e. each speaker model and the non-speech model.
- A total of 150 frames is used for a majority vote on the likelihood values to determine the classification result.
- The winning speaker model is compared with the second best model by computing the likelihood ratio.
- Since this is a good indicator of the confidence level of the decision, thresholding this value enables the detection of unknown speakers.
Downstream Applications: Meeting Browser

- E.g. access a meeting recording by clicking on the sentence of a summary
- Navigate through a transcription using a display that shows the sequence of topics that the group discussed
- Visualizations of who spoke when

Project AMI (Augmented Multiparty Interaction)
see http://www.amiproject.org/
With the increasing audio and video capture of humans in various living and working environments, there is a need to quickly understand these patterns of behavior for automated categorization.

In particular, classifying the roles that participants play in meetings is of interest.

One key element of group dynamics is dominance.

Being able to identify dominant behavior in conversational settings could potentially allow us to analyze the effectiveness of teams or to search or browse meeting data.

"Estimating the Dominant Person in Multi-Party Conversations Using Speaker Diarization Strategies"
H. Hung, Y. Huang, G. Friedland, and D. Gatica-Perez
Experiment on Dominance Estimation

Plan view of the meeting room set up. Only audio sources were used for automated dominance estimation in the cited study.
The label of the most dominant person is associated with the speaker who had the longest total speaking part at the end of each meeting.

This simple computational strategy was found to be robust, effective and fast.

More elaborate strategies are described in


Diarization system: Agglomerative clustering + fast-match
## Diarization Accuracy

<table>
<thead>
<tr>
<th>Source</th>
<th>SNR (dB)</th>
<th>Fixed number of speaker clusters</th>
<th>Automatic speaker cluster estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>KLFM</td>
<td>PCFM</td>
</tr>
<tr>
<td>MICM MH</td>
<td>31</td>
<td>27.7</td>
<td>28.1</td>
</tr>
<tr>
<td>MICM ML</td>
<td>22</td>
<td>29.4</td>
<td>28.0</td>
</tr>
<tr>
<td>SDM AT</td>
<td>21</td>
<td>34.8</td>
<td>37.8</td>
</tr>
<tr>
<td>SDM AC</td>
<td>18</td>
<td>34.1</td>
<td>33.0</td>
</tr>
<tr>
<td>speed-up: (x &gt;RT)</td>
<td>1.2</td>
<td>0.9</td>
<td>0.8</td>
</tr>
</tbody>
</table>

### Speed-up: (x >RT)

<table>
<thead>
<tr>
<th>Methods</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sources</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SNR decrease**

**Speed increase**

SCALE speaker diarization tutorial, 01/13/2010, Christian Müller
Results: Full Annotator Agreement

Note: most work on dominance estimation involves video analysis as well.
Dominance Estimation Based on Length of Contribution…

I am Gerald Friedland. My research interests focus on intelligent multimedia applications, especially technology that extracts meaning (semantic) from data created for human sensory perception. This can be visual, acoustic, or other data. In fact, I am especially interested in systems where the combination of modalities results in synergistic effects.

If you wish a physical meeting with me contact me at the address on the right. Otherwise feel free to browse through my virtual presence.

I am Kofi Boakye. A sixth-year graduate student in the Electrical Engineering and Computer Science department at the University of California, Berkeley. I work in the speech group of the International Computer Science Institute under Professor Nelson Morgan. My areas of investigation include speaker recognition and speech detection and classification in meetings. For a more detailed look at past and present work, please refer to my research page.

I am Marijn Huijbregts. My PhD research focuses on Spoken Document Retrieval (SDR). This technique makes it possible to search through audio or video the same way as it is possible to search through text documents (like search engines on the internet).

Below I have drawn a rough diagram of the procedure. First Automatic Speech Recognition (ASR) is used to translate the speech from audio or video files into text. The text and also the time in the video that it is pronounced are stored in a database. After this, when a user formulates a query, the system will search through the database and it will (hopefully) come up with some relevant audio or video fragments.

I am Chuck Wooters. My most recent publications can be found by searching the ICSI publications page.


1998: Morgan. You are all fired!

“feature kibitzer”, this statement

…now who is the most dominant speaker?

btw. those are the names and faces of people contributed with their research to this tutorial.
More Contributors

Yan Huang (fast match, prosodic features)

Oriol Vinyals (ICSIlog, prosodic features, online diarization)

Beatriz Trueba (overlap speech)

Hayley Hung (with IDIAP, dominance estimation)
Downstream Applications (Online Diarization)

Role-based Multiparty Interaction in cars

„I’m cold, please turn up the heating.“

Role-based rights model

Online Speaker Diarization

Work in progress at DFKI

SCALE speaker diarization tutorial, 01/13/2010, Christian Müller
Introduction to the agglomerative clustering approach as an example of a typical state-of-the-art diarization system.

Detection of overlap speech and run-time optimization as examples of current issues.

Incorporation of long-term (high level) features for improving accuracy.

Differences between offline and online speaker diarization

Downstream application examples