Higher-Level Features in Speaker Recognition

Winter School on Speech and Audio Processing

IIT Kanpur, January 2009

Andreas Stolcke

Speech Technology and Research Laboratory

SRI International, Menlo Park, Calif., U.S.A.

Joint work with:

E. Shriberg, L. Ferrer, S. Kajarekar, G. Tur, A. Venkataraman

Overview

- Motivation
- What are higher-level features?
- History
- Taxonomy of higher-level features
- Examples
  - Word N-gram modeling
  - State/phone/word duration modeling
  - Prosodic modeling
- Tools
  - Automatic speech recognition
  - SVM modeling
  - System combination
- Performance comparison

Motivation

- Most applied speaker recognition is based on **short-term cepstral** features
  - Cepstral features are primarily a function of speakers vocal tract shape
  - Cepstral features are affected by extraneous variables, like channel and acoustic environment
- Higher-level features aim for
  - More detail in cepstral modeling, by conditioning on additional information
  - Capturing of speaker-specific linguistic and behavioral aspects not reflected at the cepstral level

Higher-Level Features in Speaker Recognition

Terminology is imprecise, but has traditionally meant several things in the speaker recognition community:

1. Features that go beyond spectral/cepstral
2. Features that span temporal regions longer than a typical frame (10-25ms) used in cepstral analysis, often using regions of variable length
3. Features based on linguistic units, such as phones, syllables, words, or prosodic phrases.
4. Features based on automatic speech recognition (phone or word level)

History

- Early examples:
  - Pitch distribution modeling (Sonmez et al. ‘98)
  - Phone-based speaker modeling (Andrews et al. ‘01)
- “SuperSID” workshop at Johns Hopkins University, 2002
  - Explored a range of features
  - Much improvement over cepstral baseline by combining lots of systems
  - Led by MIT-LL; prosodic features and ASR provided by SRI
- Much use of high-level featured in NIST speaker recognition evaluations (SRE) in following years
  - Primary evaluation condition now used 2.5 min. of speech in train & test
  - Optional “extended data” condition with 8 x 2.5 mins of training data
  - MIT & SRI each usually had 6 or more systems in combination

History (continued)

- Recent trend has been to reduce number of high-level systems
  - To reduce computational overhead
  - Because epstral systems have gotten much better, gains from high-level features are smaller
- SRI continues to explore high-level modeling
  - Combine advances in cepstral modeling with HL features
  - Next challenge: language independent approaches

A Classification of Higher-Level Features

- We like to categorize features along following dimensions:
  - **Feature type:** what are the observations being modeled?
  - **Time span:** short (frame) versus long (or variable)
  - **ASR use** in defining **observation unit:** phone, syllable, word, phrase
  - **ASR use** in **conditioning** observation: phone, syllable, word, etc.
- Here: focus on a few feature types covering a range of levels and approaches
  - Two important additional approaches will be covered in separate lectures
- See book chapter for more complete table and references

E. E. Shriberg (2007), Higher Level Features in Speaker Recognition. In C. Müller (Ed.) *Speaker Classification I.* Volume 4343 of Lecture Notes in Computer Science / Artificial Intelligence. Springer: Heidelberg / Berlin / New York, pp. 241-259

Higher-Level Features: An overview

Feature Type	Feature Description	Time Span	ASR to Find Unit	ASR to Condition
Cepstral	phone-conditioned text-conditioned GMMs phone HMMs whole word	▪ ▪ ▪ —	Ø Ø phone, word Ø	phone word, syll. phone N-gram
Cepstral-Derived	MLLR adapt. transforms	▪	word, unc. phone	phone
Acoustic Tokenization	phone N-gram freq. conditioned pron. model	— —	unc. phone unc. phone	Ø phones
Prosodic	dynamics duration syllable-pros. sequences	— — —	Ø state, phone, syllable	Ø phone, word word
Lexical	word N-grams	—	word	Ø

Example Features and Models

Disclaimer on Results

- Many of the results presented are historical
- Results obtained on different training/test sets
- Baselines vary and get better the more recent the results
- Gains over baseline may also vary
  - The better the baseline, the less typically the gain

- **Your mileage may vary !**

Word N-gram Modeling

Word N-gram Features

- Idea (Doddington 2001):
  - Word usage can be idiosyncratic to a speaker
  - Model speakers by relative frequencies of word N-grams
  - Reflects vocabulary AND grammar
  - Cf. similar approaches for authorship and plagiarism detection on text documents.
  - First (unpublished) use in speaker recognition: Heck et al. (1998)
- Implementation:
  - Get 1-best word recognition output
  - Extract N-gram frequencies
  - Model likelihood ratio OR
  - Model frequency vectors by SVM

I_shall	0.002
I_think	0.025
I_would	0.012
…	…

Word N-gram Modeling: Likelihood ratios

- Model N-gram token log likelihood ratio
- Numerator: speaker language model estimated from enrollment data
- Denominator: background language model estimated from large speaker population
- Normalize by token count

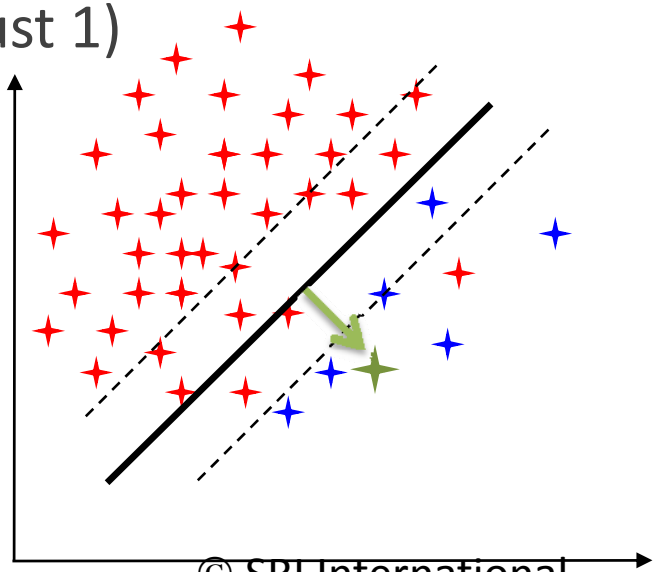
$$Score = \frac{\sum_j \log \frac{\Lambda_{Speaker}(j)}{\Lambda_{Background}(j)}}{\sum_j 1}$$

- Choose all reasonably frequent bigrams or trigrams, or a weighted combination of both

Speaker Recognition with SVMs

- Each speech sample (training or test) generates a point in a derived feature space
- The SVM is trained to separate the target sample from the impostor (= UBM) samples
- Scores are computed as the Euclidean distance from the decision hyperplane to the test sample point
- SVMs training is biased against misclassifying positive examples (typically very few, often just 1)

Background sample
Target sample
Test sample



Feature Transforms for SVMs

- SVMs have been a boon for higher-level (as well as cepstral speaker recognition) research – they allow great flexibility in the choice of features
- However, we need a "sequence kernel"
- Dominant approach: transform variable-length feature stream into fixed, finite-dimensional feature space
- Then use linear kernel
- All the action is in the feature transform!

- We will discuss more interesting feature transforms in the 2nd and 3rd lectures!

Word N-gram Modeling with SVMs

- Features: relative word N-gram frequencies of unigrams, bigrams, and trigrams

I_shall	0.002
I_think	0.025
I_would	0.012
…	…

- Note: features subject to ASR error
- Feature selection: by frequency on background training data (about top 100k most frequent N-grams)
- Since enrollment and test data is short (compared to background data), most feature values are zero
  - SVM software should be optimized for sparse feature vectors!
- Feature scaling and normalization: see tomorrow's lecture

Word N-gram Modeling: Results

- Results obtained on SRE'04 test data (EER)
- Baseline system: cepstral GMM
- Score-level combiner: neural network

	1-side training	8-side training
Ceptral system	11.27	6.54
Word ngrams (LLR)	27.81	16.36
Word ngrams (SVM)	23.06	12.36
Cepstral + Word ngrams (SVM)	10.03	3.27
Relative improvement	11%	50%

- Conclusions:
  - SVM modeling substantially better than LLR
  - Word N-grams by themselves are not competitive with baseline, but
  - Combination with cepstral baseline yields significant gains

Duration-conditioned Word N-grams

- Most frequent 5000 words are binned into two categories, “slow” and “fast”, with respect to their duration.
- Then, each of word w is labeled as either w_{slow} or w_{fast} while computing the N-gram frequencies.
- Less frequent words are treated as before (duration-independently)
- The background set comprised 1971 conversation sides from the Fisher corpus, Switchboard-2 NIST SRE 2003 data, Switchboard-2 Phase 5 data.
- The values are then rank-normalized to the range [0;1], using the background data as the reference distribution.
- Details see Tur et al. (2007)

Duration-conditioned Word N-grams: Results

- Results on SRE'06 test data (EER)

	1-side training	8-side training
Standard Word N-grams	26.53	11.14
Duration-conditioned	23.46	9.95
Relative improvement	8.5%	10.7%

- Note: similar approach based on *pronunciation-specific* word labels was not as successful.

Duration Modeling

Duration Modeling

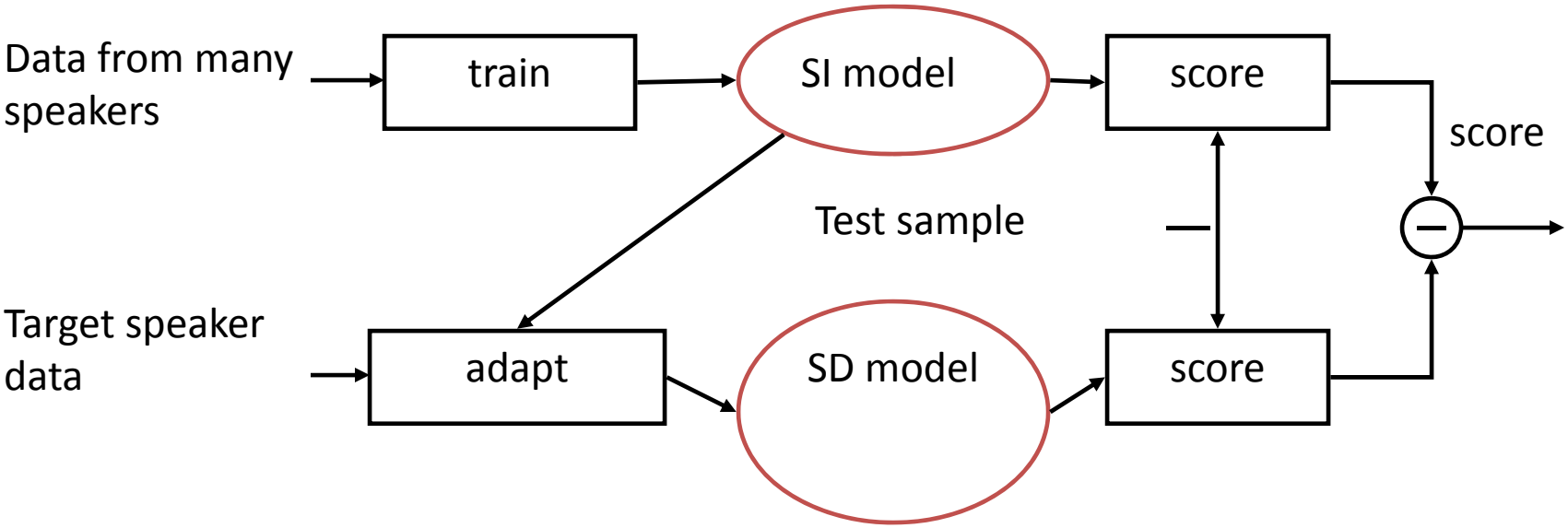
- **Goal:** capture speaker-specific duration patterns for particular words or phones
- Each word (or phone) is represented by a vector comprised of the durations of the individual phones (or states) it contains. Example:

"that"

- Gadde (2000) successfully used duration features for speech recognition
- Here, we investigate and extend duration features for the task of speaker recognition

Duration Modeling (cont.)

- Vectors modeled by Gaussian Mixture Models.
- Speaker-dep. models obtained through adaptation of a SI model trained on data from many speakers.
- SD models then used to score test samples. Score normalized by score obtained using SI model.



Duration Features

- 3 types of features:
  - **Phone-in-word features:** Sequence of phone durations in word. Number of components depends on pronunciation.
    E.g., w:***that*** **dh+ax+t** → (**4 8 6**)
  - **Phone features:** Duration of phone. Single-component vectors.
    E.g., p:**dh** → (**4**) p:**ax** → (**8**) p:**t** → (**6**)
  - **State-in-phone:** Sequence of state durations in phone. Three-component vectors.
    E.g., s:**dh** → (**2 1 1**) s:**ax** → (**3 2 3**) s:**t** → (**1 1 4**)
- Obtain features from either forced alignments to true words, or to recognized words.

Duration Model Training and Adaptation

- Train speaker independent GMMs for each word and each phone (one component and three component models)
- Then obtain the SD models through MAP adaptation of the SI model
- Adapt means and weights. Weight is based on number of speaker dependent samples available
- The SI model size has to be such that during adaptation, most of the Gaussians have some number of speaker dependent samples to be adapted to

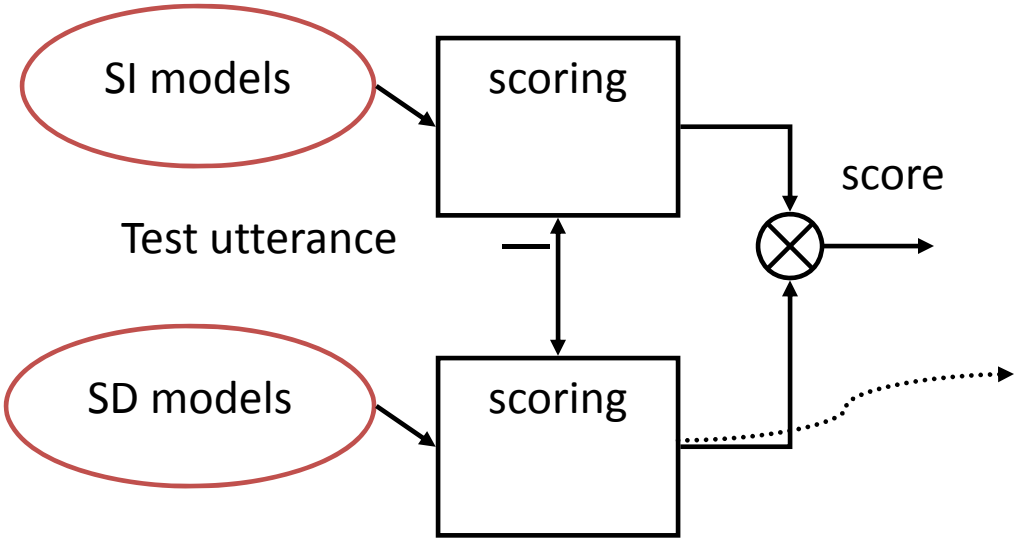
Duration Model Training and Adaptation (cont.)

- Duration patterns often change when the speaker is about to make a pause
- We therefore condition models on the context:

1. ***Pause context*** models are trained using the samples that are found before a pause longer than 200msec.
2. ***Word context*** models are trained using all other samples (no following pause).

Duration Scoring Procedure

- Three separate scores are obtained, one for each set of models: phone-in-word, phones, state-in-phone.
- Compute each score as the sum of the log-likelihoods of the feature vectors in the test utterance given their SD models, divided by the number of components scored and normalized by the score obtained using the SI model.



P-in-W, P, S-in-P features

$$\frac{\sum_{x} \log\left(p(x/\text{model\_for\_x})\right)}{\text{num\_comp\_scored}}$$

Duration Scoring Procedure (cont.)

- **Back-off strategy:** when context dependent model not adapted to speaker with more than a certain number of samples, use context independent model to score instead.
- **Avoiding non-robustly adapted models:** Score only those models that were adapted to the speaker with more than 5 samples, to avoid non-robust models.

System Combination

- Duration systems were combined with a GMM standard system (from 2003) that uses Mel-frequency cepstral coefficients as features
- To assess whether duration features complement lexical information, also combined with word bigram feature system (Doddington 2001)
- Combination results obtained using multilayer perceptron with one hidden layer with 10 nodes.
- Training/test database: NIST SRE'01 (Switchboard 1)
- Used N-fold jack-knifing to train the classifiers.

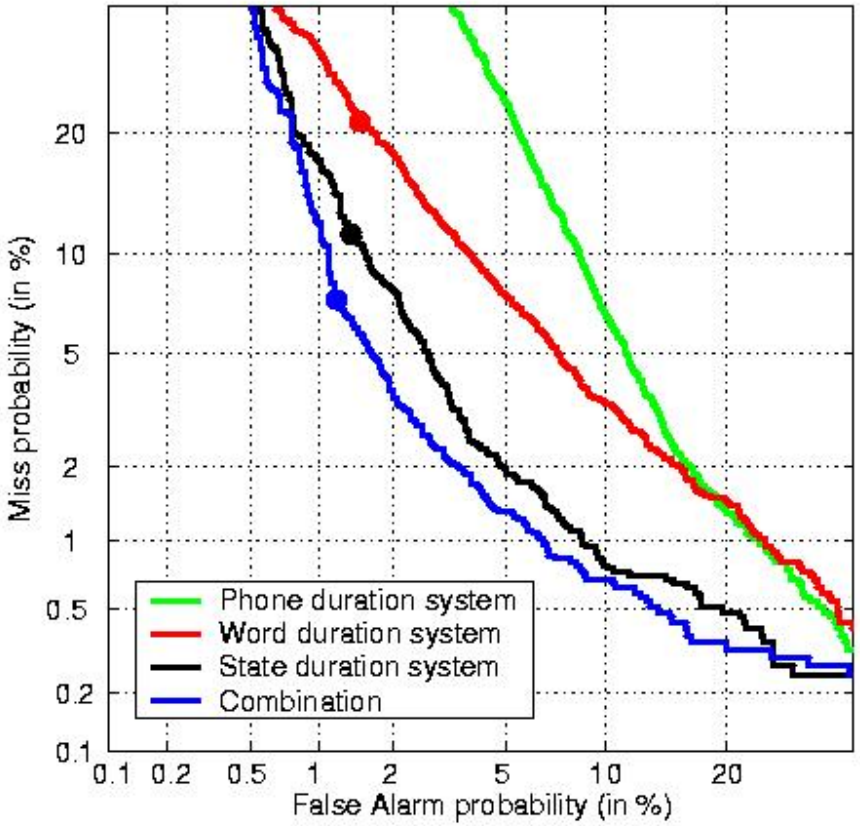
Duration Modeling: Results

	EER on true transc.	EER on rec. words
Baseline	0.90 %	
Word bigrams	8.65 %	9.30 %
State-in-phone durations	3.71 %	3.30 %
Phone durations	10.88 %	8.82 %
Phone-in-word durations	5.22 %	6.22 %

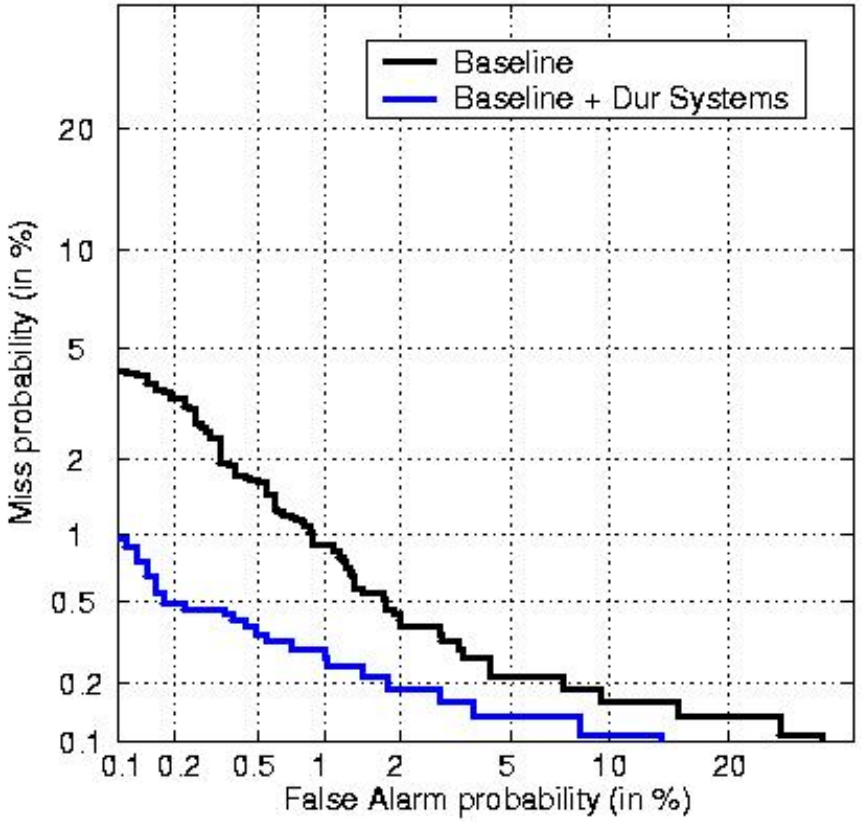
- Phone-in-word duration models and word N-gram models degrade when using recognized words, but state-in-phone and phone duration models improve
- Speaker-specific misrecognitions benefit these systems ?

Duration Modeling: DET Curves

Duration systems and combination (for rec. words)



Combination of baseline and duration systems



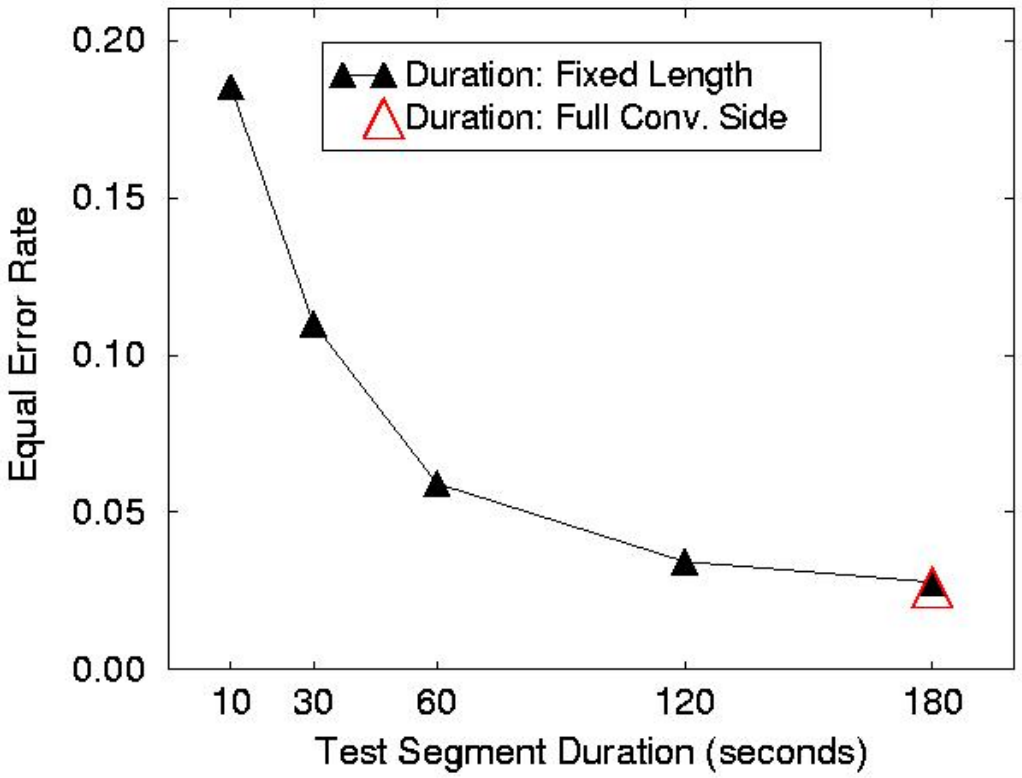
Duration System Combined with Baseline

- Adding the duration features both to the baseline alone and to the baseline with lexical features reduces the EER by 50%.

	EER, rec. words
Baseline	0.90 %
All duration systems	2.59 %
Baseline + all duration	0.40 %
Baseline + word ngram	0.57 %
Baseline + all duration + word ngram	0.29 %

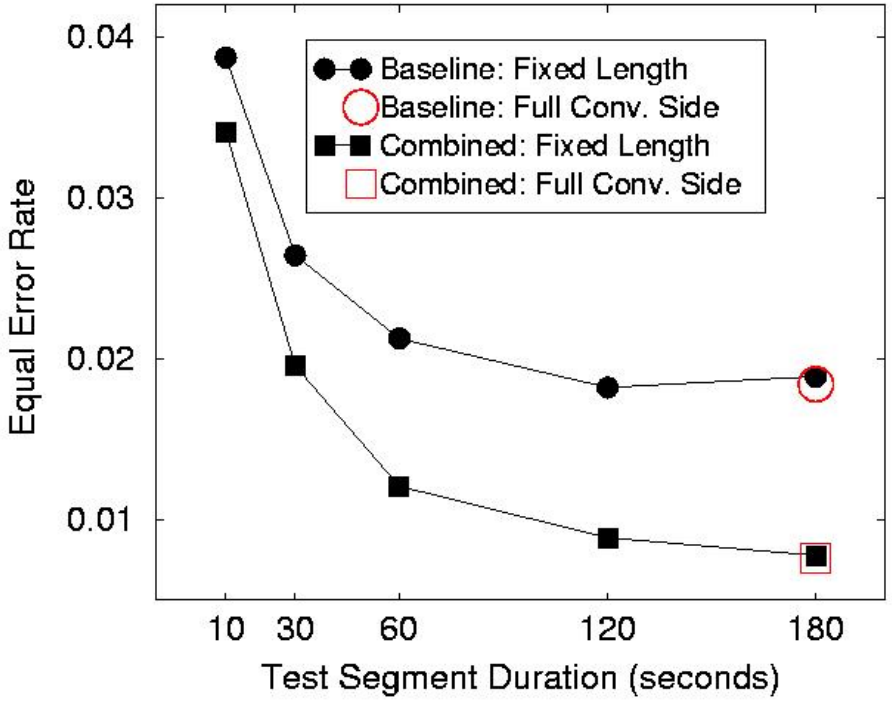
Duration Performance with Varying Test Length

- Create shorter tests by concatenation of speech segments with only small pauses embedded.
- For each conversation side-length test we now have several shorter tests.



Baseline and Combination for Varying Test Length

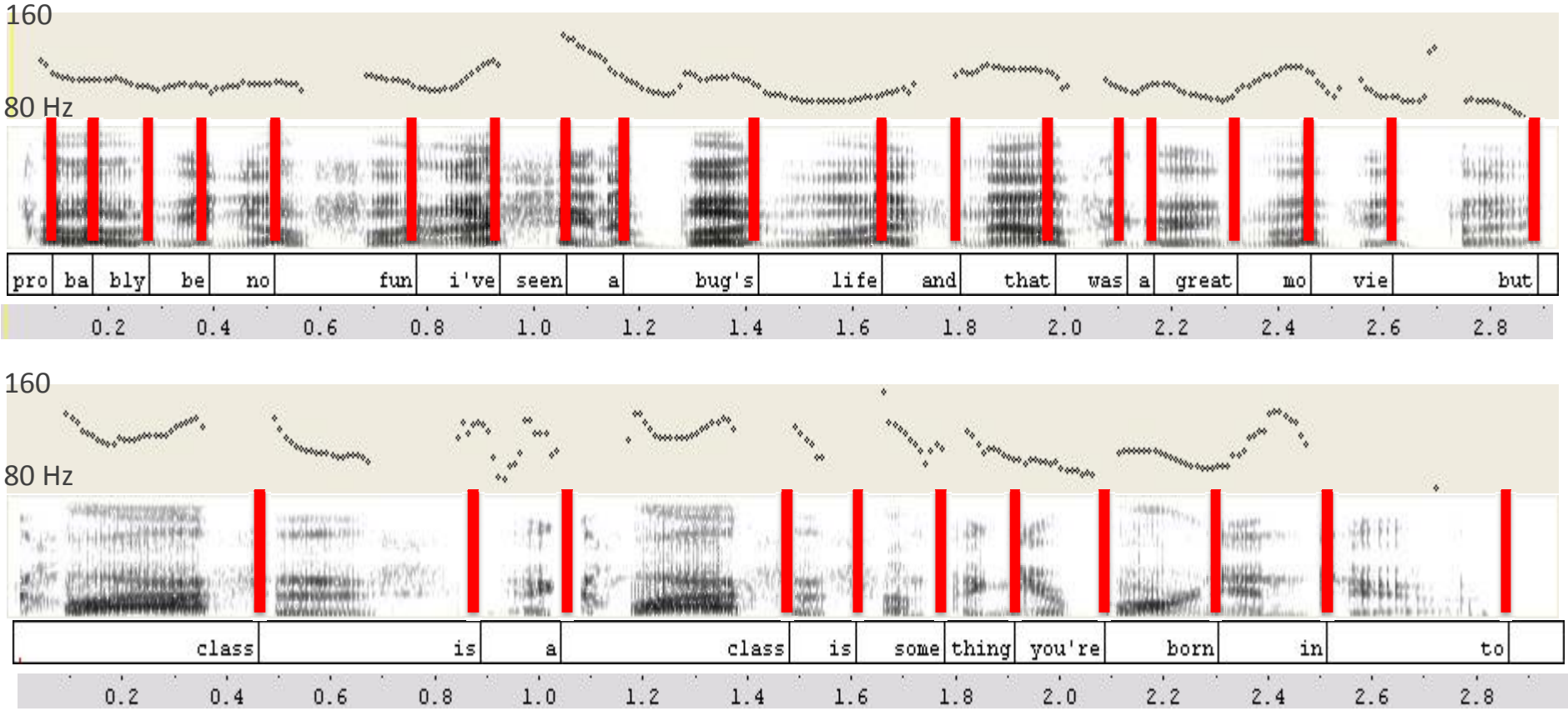
- Even at short test lengths, duration models give an improvement.
- Baseline seems to level off at 2 min of test data, while duration models do not.
- Contribution of duration increases with test length.



Prosodic Modeling

Prosodic Modeling: Motivation

- Two male speakers confused by SRI SRE'08 cepstral system
- Very similar pitch range. Same elapsed time shown for each
- But: 1st speaker has nearly twice the word/syllable rate as 2nd



Prosodic Modeling: History

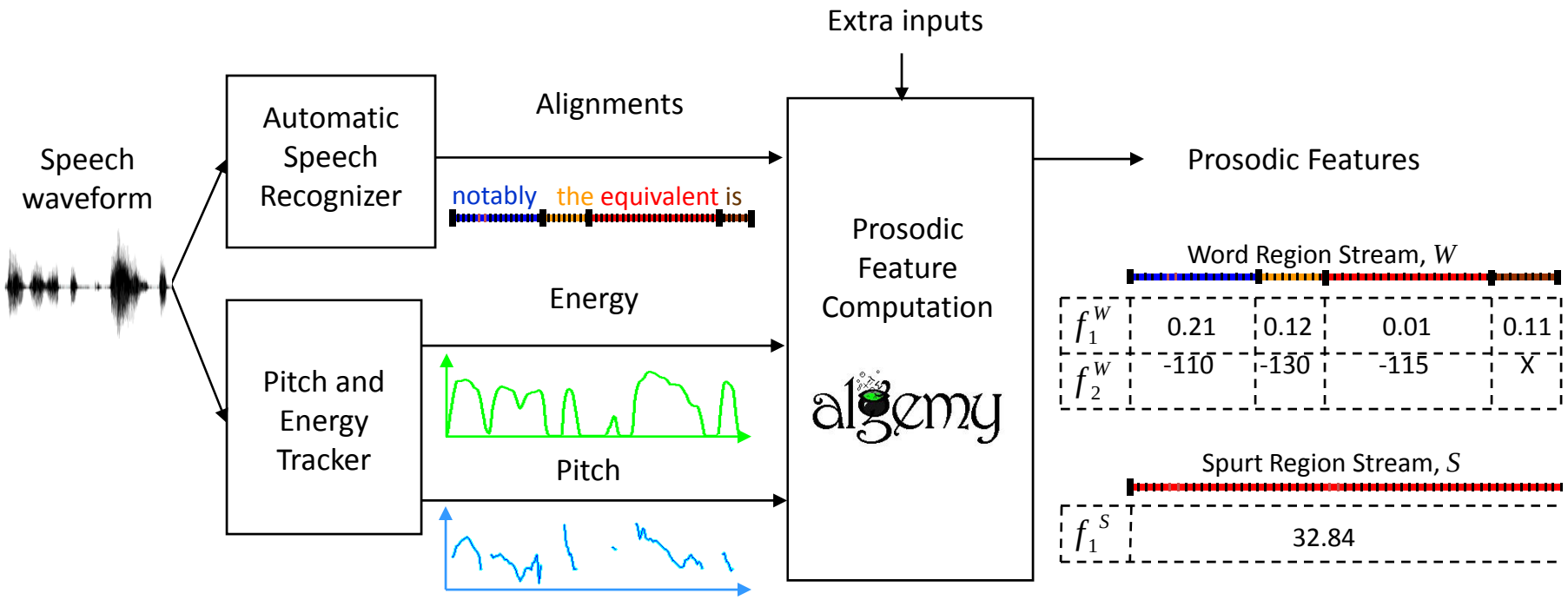
- Early work: frame-level model pitch distribution (Sonmez et al. '98), no sequence modeling
- Simple pitch and energy dynamics model based on discretized features and bigram modeling (Adami et al. '03)
- NERFs: Non-uniform extraction region features (Kajarekar et al. '04)
  - Exract prosodic features from longer regions, e.g., between pauses
- SNERFs: Syllable-based NERFs (Shriberg et al. '05)
  - Extract prosodic features for each syllable, model sequences
- GNERFs: Grammar-based SNERFs (Shriberg & Ferrer '07)
  - Condition syllable-based prosodic features on word identity and and grammatical word class

Next slides: explain the last three approaches, developed at SRI

Prosody Modeling at SRI

- NERF = Non-uniform extraction region features
- Goal: model the prosodic characteristics of the speaker's speech
- Yield best performance of all of SRI's "stylistic" (non-cepstral) systems
- Also, yields the most gains when combined with state-of-the-art cepstral models

Prosodic NERF Extraction

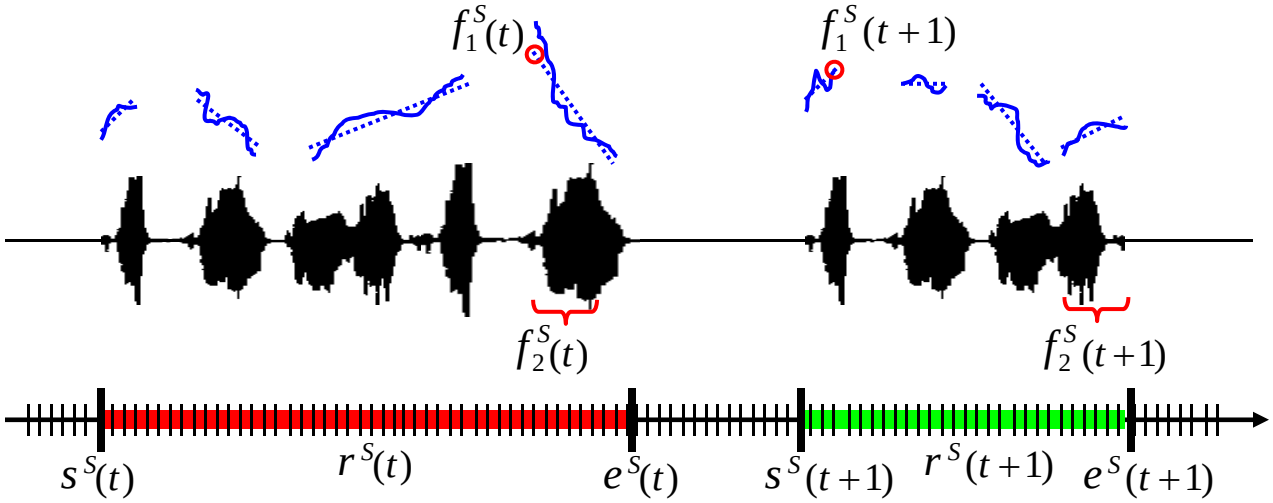


- Spert = region of speech delimited by pauses > 0.5 seconds

Prosodic Feature Example: Spurt NERFs

Pitch and stylized pitch

Waveform



S: *Spurt* region stream

f_1^S: Maximum stylized pitch	$f_1^S(t)$	$f_1^S(t+1)$
f_2^S: Duration of last vowel	$f_2^S(t)$	$f_2^S(t+1)$

SNERFs: Syllable-based NERFs

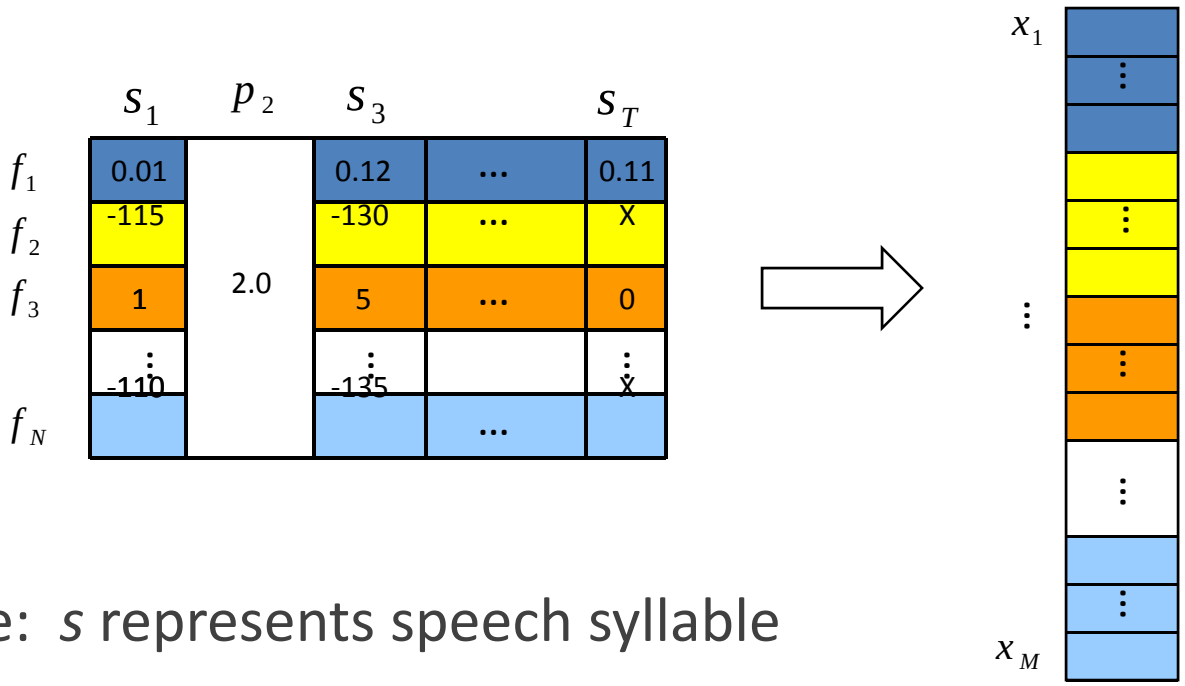
- Currently using *syllables* as regions
  - Syllables determined automatically from recognition output
  - … and phonetic syllabification rules (NIST's tsylb2 software)
- Hundreds of pitch, energy and duration features
  - Features are frequently undefined, are highly correlated and have continuous, discrete, or mixed distributions
  - Currently computing 140 features

GNERFs: Grammar NERFs

- Basic regions are again the syllables
- Same features, but extracted only over specific “wordlists”
- Each wordlist contains a list of constrains
- Each constrain consists of
  - a specific word,
  - a specific part of speech (POS) tag,
  - a word+POS tag pair
- Example: backchannels
  - *yeah, yes, ok, uhhuh, oh, …*

SVM Feature Transformation

- Need to transform messy variable-length SNERF stream into a single continuous-valued, fixed-length vector



- Note: *s* represents speech syllable

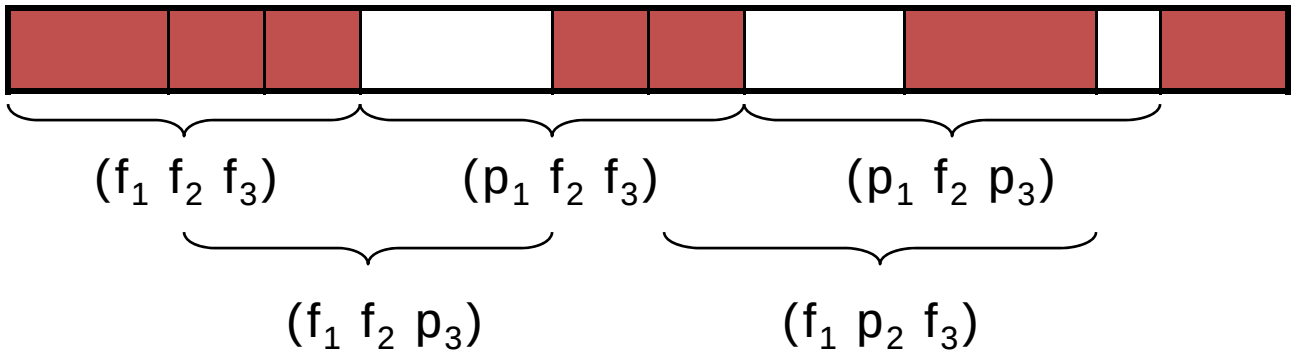
  p represents non-speech pause

SVM Feature Transformation (cont.)

Transform one feature at a time and concatenate the results

For each feature:

- Create GMM models for each unigram, 2gram, 3gram, 4gram
- For each N-gram length include several models with pauses in different slots (*tokens*)
- Trigram example:



SVM Feature Transformation (cont.)

- For each token
  - Train a GMM from held-out data
  - The transformed features consists of the *posterior probability of each Gaussian given the data*
- Finally,
  - Concatenate transformed features for all tokens
  - Rank-normalize each component
  - Take this vector as input to the SVM

Syllable-based Prosodic Modeling: Results

- Results on SRE'06 database (EER)
- Cepstral baseline: MLLR-SVM system (see 3rd lecture)
- Best systems use intersession-session (intra-speaker) variability compensation (ISV, see 3rd lecture)

	1-side training	8-side training
SNERFs	12.08	5.42
SNERFs+GNERFs	11.54	5.37
SNERFs+GNERFs (ISV)	10.41	3.73
MLLR	3.99	2.14
MLLR + SNERFs+GNERFs	3.72	1.69

~ 21% improvement !!!

Prosodic Modeling: Another Approach

- NERFs require speech recognition for pause detection, syllabification, and word conditioning
- Alternative approach that does not require ASR (Dehak et al. '07):
  - model raw energy and pitch tracks by fitting Legendre polynomials
  - Polynomial coefficients are features
- Two modeling approaches:
  - GMM supervector (with factor analysis for ISV compensation) (Dehak et al. '07)
  - GMM weight transforms (with nuisance attribute projection for ISV compensation) (Ferrer et al. '07)
- SRI's 2008 NIST SRE system incorporated both approaches, as well as ASR-dependent (SNERFs+GNERFs)

Results Comparison

- Results (EER) on SRE'08 English dataset
- All systems use ISV compensation
- Phone duration system was dropped

Systems (gray = ASR-dependent)	1-side training	8-side training
Cepstral GMM	2.914	1.277
Prosodic w/ASR	10.016	3.502
State-in-phone Durations	14.820	9.208
Prosodic w/o ASR (poly)	17.180	10.253
Prosodic w/o ASR (supervector)	17.765	12.282
Phone-in-word durations	19.626	8.113
Word N-gram	20.685	7.714

- No combination results for just these systems

Summary

- Higher-level feature capture aspects of speech complementary to cepstral features
  - Linguistic units
  - Longer-term patterns
  - Stylistic aspects, as opposed to vocal-tract shape
- Showed examples from three feature domains:
  - Word N-grams
  - Durations of sub-word units
  - Prosodic features (pitch, energy, durations)
- SVM modeling is a key tool, enabled by suitable feature transforms
- Found substantial gains in combination with cepstral baseline system in each case

Thank you – Questions?

References (1)

A. G. Adami, R. Mihaescu, D. A. Reynolds, and J. J. Godfrey (2003), Modeling Prosodic Dynamics for Speaker Recognition, *Proc. IEEE ICASSP*, vol. 4, pp. 788-791, Hong Kong.

W. D. Andrews, M. A. Kohler, and J. P. Campbell (2001), Phonetic Speaker Recognition, *Proc. Eurospeech, pp. 149–153, Aalborg.*

B. Baker, R. Vogt, and S. Sridharan (2005), Gaussian Mixture Modelling of Broad Phonetic and Syllabic Events for Text-Independent SpeakerVerification, *Proc. Eurospeech*, pp. 2429–2432, Lisbon.

K. Boakye and B. Peskin (2004), Text-Constrained Speaker Recognition on a Text-Independent Task, *Proc. Odyssey Speaker and Language Recognition Workshop*, pp. 129-134, Toledo, Spain.

T. Bocklet and E. Shriberg (2009), Speaker Recognition Using Syllable-Based Constraints for Cepstral Frame Selection, *Proc. IEEE ICASSP*, Taipei, to appear.

W. M. Campbell (2002), Generalized Linear Discriminant Sequence Kernels for Speaker Recognition, *Proc. IEEE ICASSP*, vol. 1, pp. 161-164, Orlando, FL.

W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek (2004a), Phonetic Speaker Recognition with Support Vector Machines, in *Advances in Neural Processing Systems 16*, pp. 1377-1384, MIT Press, Cambridge, MA.

W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek (2004b), High-level speaker verification with support vector machines, *Proc. IEEE ICASSP*, vol. 1, pp. 73-76, Montreal.

W. M. Campbell, D. E. Sturim, D. A. Reynolds (2006), Support vector machines using GMM supervectors for speaker verification, *IEEE Signal Proc. Letters* 13(5), 308-311.

N. Dehak, P. Dumouchel, and P. Kenny (2007), Modeling Prosodic Features With Joint Factor Analysis for Speaker Verification, *IEEE Trans. Audio Speech Lang. Proc.* 15(7), 2095-2103.

G. Doddington (2001), Speaker Recognition based on Idiolectal Differences between Speakers, *Proc. Eurospeech*, pp. 2521-2524, Aalborg.

References (2)

M. Ferras, C. C. Leung, C. Barras, and J.-L. Gauvain (2007), Constrained MLLR for Speaker Recognition, *Proc. IEEE ICASSP*, vol. 4, pp. 53-56, Honolulu.

L. Ferrer, E. Shriberg, S. Kajarekar, and K. Sonmez (2007), Parameterization of Prosodic Feature Distributions for SVM Modeling in Speaker Recognition, *Proc. IEEE ICASSP*, vol. 4, pp. 233-236, Honolulu, Hawaii.

L. Ferrer, K. Sonmez, and E. Shriberg (2008a), An Anticorrelation Kernel for Improved System Combination in Speaker Verification. *Proc. Odyssey Speaker and Language Recognition Workshop*, Stellenbosch, South Africa.

L. Ferrer, M. Graciarena, A. Zymnis, and E. Shriberg (2008b), System Combination Using Auxiliary Information for Speaker Verification, *Proc. IEEE ICASSP*, pp. 4853-4857, Las Vegas.

L. Ferrer (2008), Modeling Prior Belief for Speaker Verification SVM Systems, *Proc. Interspeech*, pp. 1385-1388, Brisbane, Australia.

V. R. R. Gadde (2000), Modeling word duration, Proc. ICSLP, pp. 601-604, Beijing.

A. O. Hatch, B. Peskin, and A. Stolcke (2005a), Improved Phonetic Speaker Recognition using Lattice Decoding, *Proc. IEEE ICASSP*, vol. 1, pp. 169-172, Philadelphia.

A. O. Hatch, A. Stolcke, and B. Peskin (2005b), Combining Feature Sets with Support Vector Machines: Application to Speaker Recognition. *Proc. IEEE Speech Recognition and Understanding Workshop*, pp. 75-79, San Juan, Puerto Rico.

L. Heck et al. (1998), SRI System Description, NIST SRE-98 evaluation.

S. Kajarekar, L. Ferrer, K. Sonmez, J. Zheng, E. Shriberg, and A. Stolcke (2004), Modeling NERFs for Speaker Recogniition, *Proc. Odyssey Speaker Recognition Workshop*, pp. 51-56, Toledo, Spain.

S. S. Kajarekar (2005), Four Weightings and a Fusion: A Cepstral-SVM System for Speaker Recognition. *Proc. IEEE Speech Recognition and Understanding Workshop*, pp. 17-22, San Juan, Puerto Rico.

Z. N. Karam and W. M. Campbell (2008), A Multi-class MLLR Kernel for SVM Speaker Recognition, *Proc. IEEE ICASSP* pp. 4117-4120, Las Vegas.

References (3)

P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel (2005), Factor Analysis Simplified, *Proc. IEEE ICASSP*, vol. 1, pp. 637-640, Philadelphia.

P. Kenny, G. Boulianne, P.Ouellet, and P. Dumouchel (2006), Improvements in Factor Analysis Based Speaker Verification, *Proc. IEEE ICASSP*, vol. 1, pp. 113-116, Toulouse.

D. Klusacek, J. Navrátil, D. A. Reynolds, and J. P. Campbell (2003), Conditional pronunciation modeling in speaker detection, *Proc. IEEE ICASSP*, vol. 4, pp. 804-807, Hong Kong.

J. Navrátil, Q. Jin, W. D. Andrews, and J. P. Campbell (2003), Phonetic Speaker Recognition Using Maximum-Likelihood Binary-Decision Tree Models, *Proc. IEEE ICASSP*, vol. 4, pp. 796-799, Hong Kong.

A. Park and T. J. Hazen (2002), ASR Dependent Techniques for Speaker Identification, *Proc. ICSLP*, pp. 1337–1340, Denver.

D. A. Reynolds, T. F. Quatieri, and R. B. Dunn (2000), Speaker Verification Using Adapted Gaussian Mixture Models, *Digital Signal Processing* 10, *181-202.*

D. Reynolds (2003), Channel Robust Speaker Verification via Feature Mapping, *Proc. IEEE ICASSP*, vol. 2, pp. 53-56, Hong Kong.

E. Shriberg, L. Ferrer, S. Kajarekar, A. Venkataraman, and A. Stolcke (2005), Modeling prosodic feature sequences for speaker recognition, *Speech Communication* 46(3-4), 455-472.

E. E. Shriberg (2007), Higher Level Features in Speaker Recognition, in C. Müller (Ed.) *Speaker Classification I.* Volume 4343 of Lecture Notes in Computer Science / Artificial Intelligence. Springer: Heidelberg / Berlin / New York, pp. 241-259.

E. Shriberg and L. Ferrer (2007), A Text-Constrained Prosodic System for Speaker Verification, *Proc. Eurospeech*, pp. 1226-1229, Antwerp.

E. Shriberg, L. Ferrer, S. Kajarekar, N. Scheffer, A. Stolcke, and M. Akbacak (2008), Detecting Nonnative Speech Using Speaker Recognition Approaches. *Proc. Odyssey Speaker and Language Recognition Workshop*, Stellenbosch, South Africa.

References (4)

A. Solomonoff, C. Quillen, and I. Boardman (2004), Channel Compensation for SVM Speaker Recognition, *Proc. Odyssey Speaker and Language Recognition Workshop*, pp. 57-62, Toledo, Spain.

K. Sonmez, E. Shriberg, L. Heck, and M. Weintraub (1998), Modeling Dynamic Prosodic Variation for Speaker Verification, *Proc. ICSLP*, pp. 3189-3192, Sydney.

A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, and A. Venkataraman (2005), MLLR Transforms as Features in Speaker Recognition, *Proc. Eurospeech*, pp. 2425-2428, Lisbon.

A. Stolcke, S. Kajarekar, L. Ferrer, and E. Shriberg (2007), Speaker Recognition with Session Variability Normalization Based on MLLR Adaptation Transforms, *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7), 1987-1998.

A. Stolcke and S. Kajarekar (2008), Recognizing Arabic Speakers with English Phones. *Proc. Odyssey Speaker and Language Recognition Workshop*, Stellenbosch, South Africa.

A. Stolcke, S. Kajarekar, and L. Ferrer (2008), Nonparametric Feature Normalization for SVM-based Speaker Verification, *Proc. IEEE ICASSP*, pp. 1577-1580, Las Vegas.

D. E. Sturim, D. A. Reynolds, R. B. Dunn, and T. F. Quatieri (2002), Speaker Verification Using Text-Constrained Gaussian Mixture Models, *Proc. IEEE ICASSP*, vol. 1, pp. 677-680, Orlando.

G. Tur, E. Shriberg, A. Stolcke, and S. Kajarekar (2007), Duration and Pronunciation Conditioned Lexical Modeling for Speaker Recognition, *Proc. Eurospeech*, pp. 2049-2052, Antwerp.

R. Vogt, B. Baker, and S. Sridharan (2005), Modelling Session Variability in Text-independent Speaker Verification, *Proc. Eurospeech*, pp. 3117-3120, Lisbon.

M. A. Zissman and E. Singer (1994), Automatic language identification of telephone speech messages using phoneme recognition and N-gram modeling, *Proc. IEEE ICASSP*, vol. 1, pp. 305-308, Adelaide.