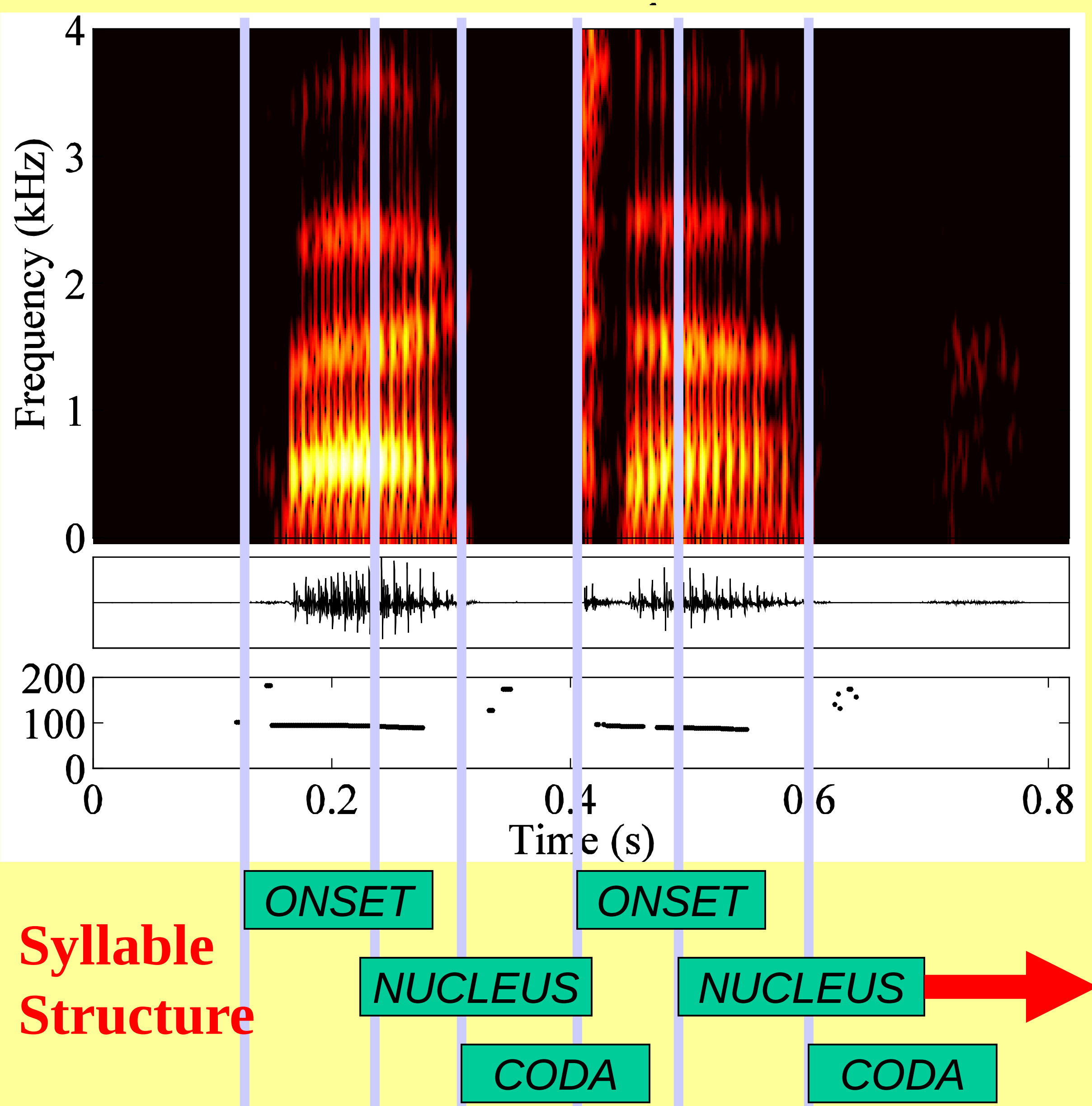
Landmark-Based Speech Recognition: Status Report, 7/21/2004

Status Report: Outline

1. Review of the paradigm
2. Experiments that have been used in rescoring
   - SVM: training on Switchboard vs. NTIMIT
   - Acoustic features: MFCCs vs. rate-scale
   - Training the pronunciation model
   - Event-based smoothing with, w/o pronunciation model
   - Results for one talker in RT03-devel
3. Ongoing experiments: Acoustic modeling
4. Ongoing experiments: Pronunciation modeling
5. Ongoing experiments: Rescoring methods

1. Landmark-Based Speech Recognition



Lattice hypothesis:
… backed up …

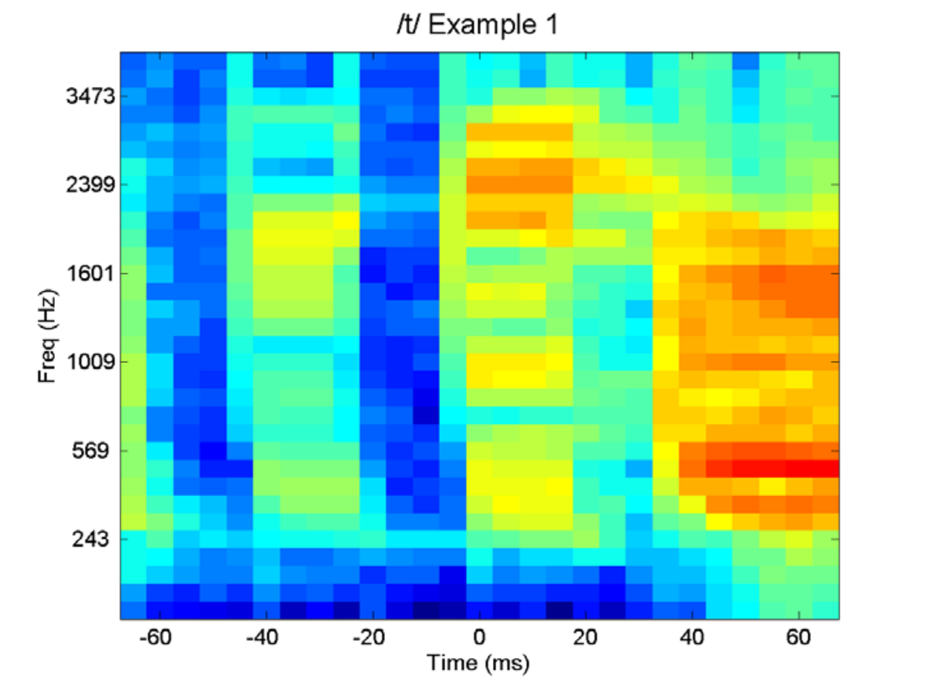
Words
Times

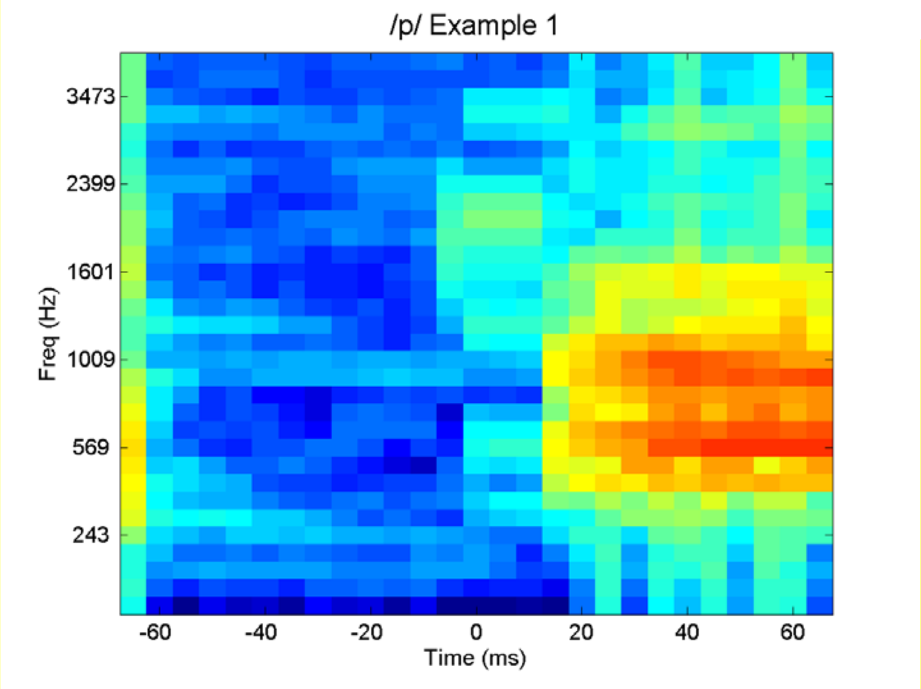
Scores

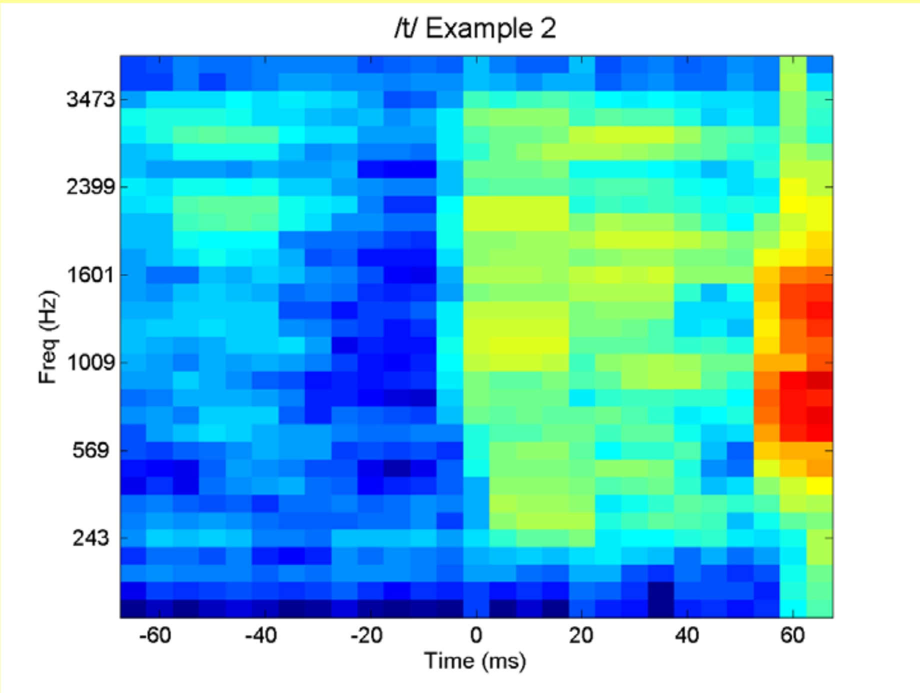
Pronunciation Variants:

… backed up …
… backtup ..
… back up …
… backt ihp …
… wackt ihp…
…

Acoustic Feature Vector: A Spectrogram Snapshot (plus formants and auditory features)







Two types of SVMs: landmark detectors (p(landmark(t)), landmark classifiers (p(place-features(t)|landmark(t))

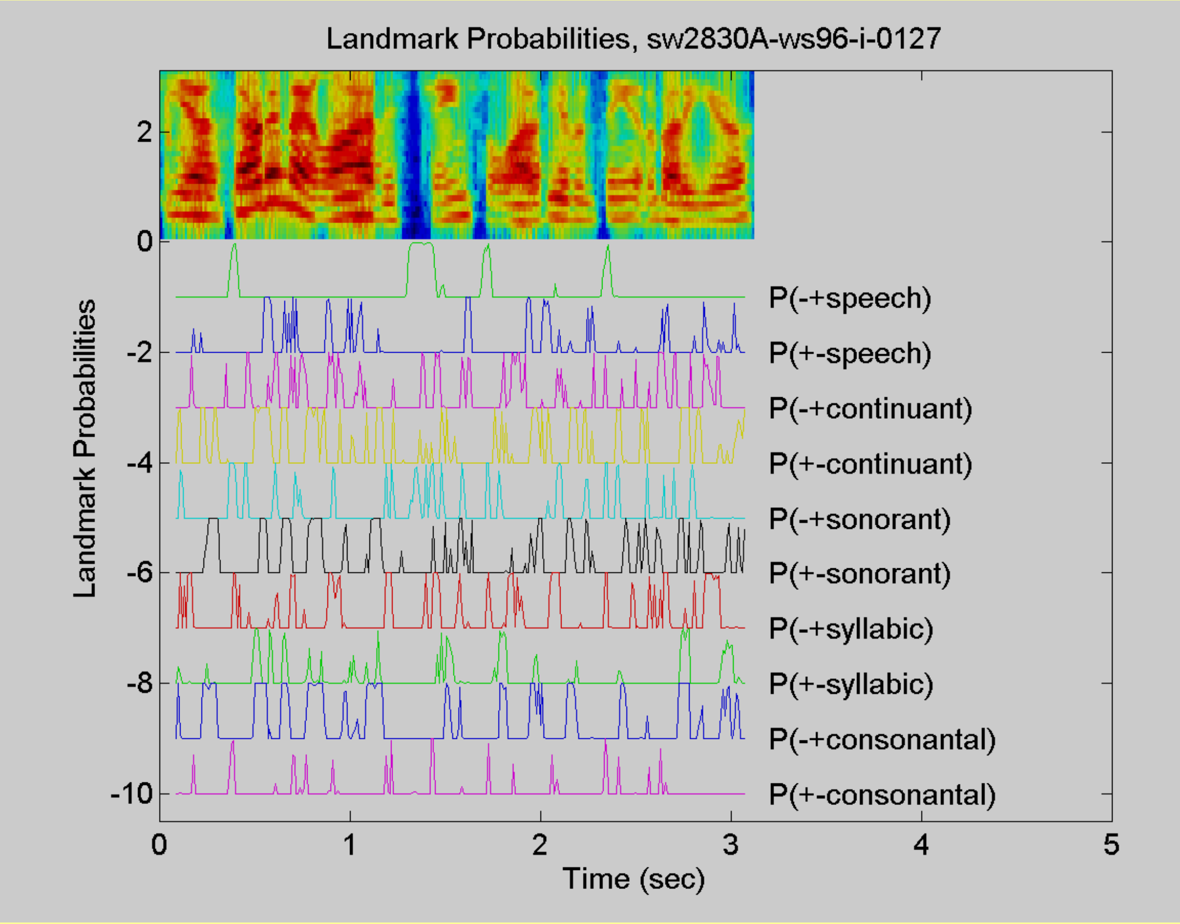
2000-dimensional acoustic feature vector

SVM

Discriminant $y_i(t)$

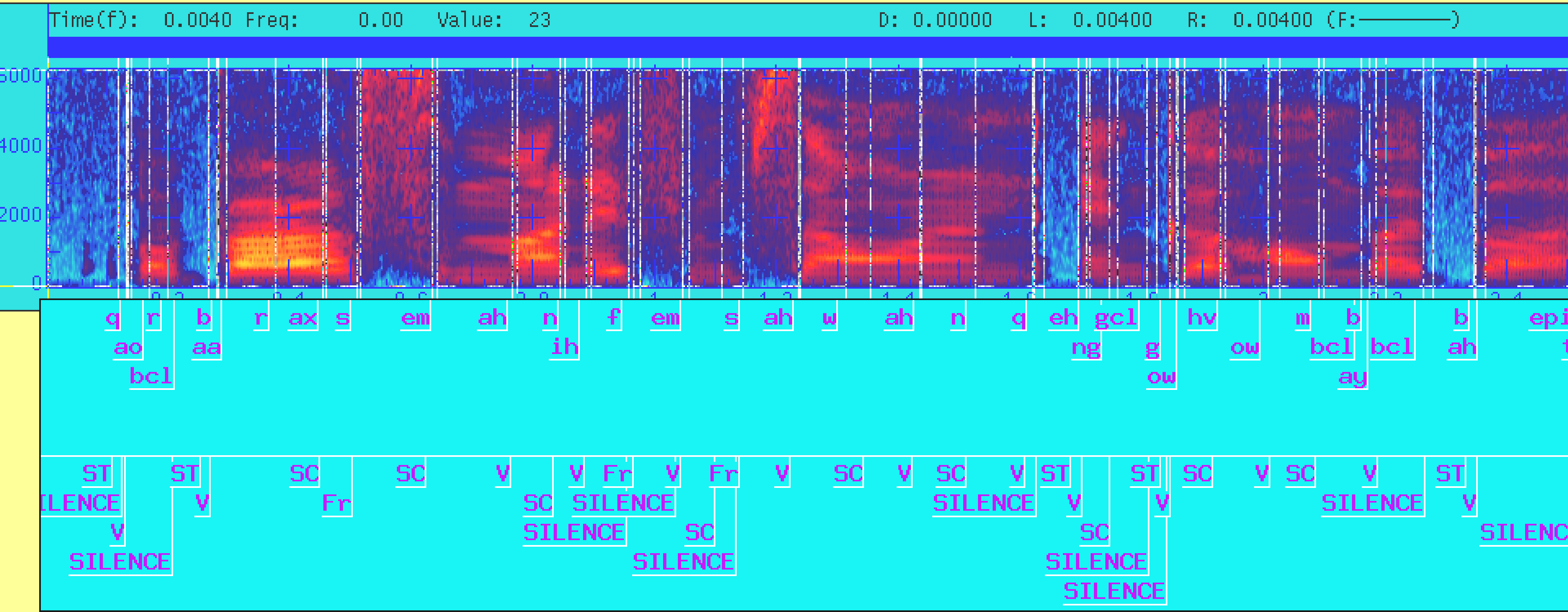
Sigmoid or Histogram

Posterior probability of distinctive feature $p(d_i(t)=1 \mid y_i(t))$

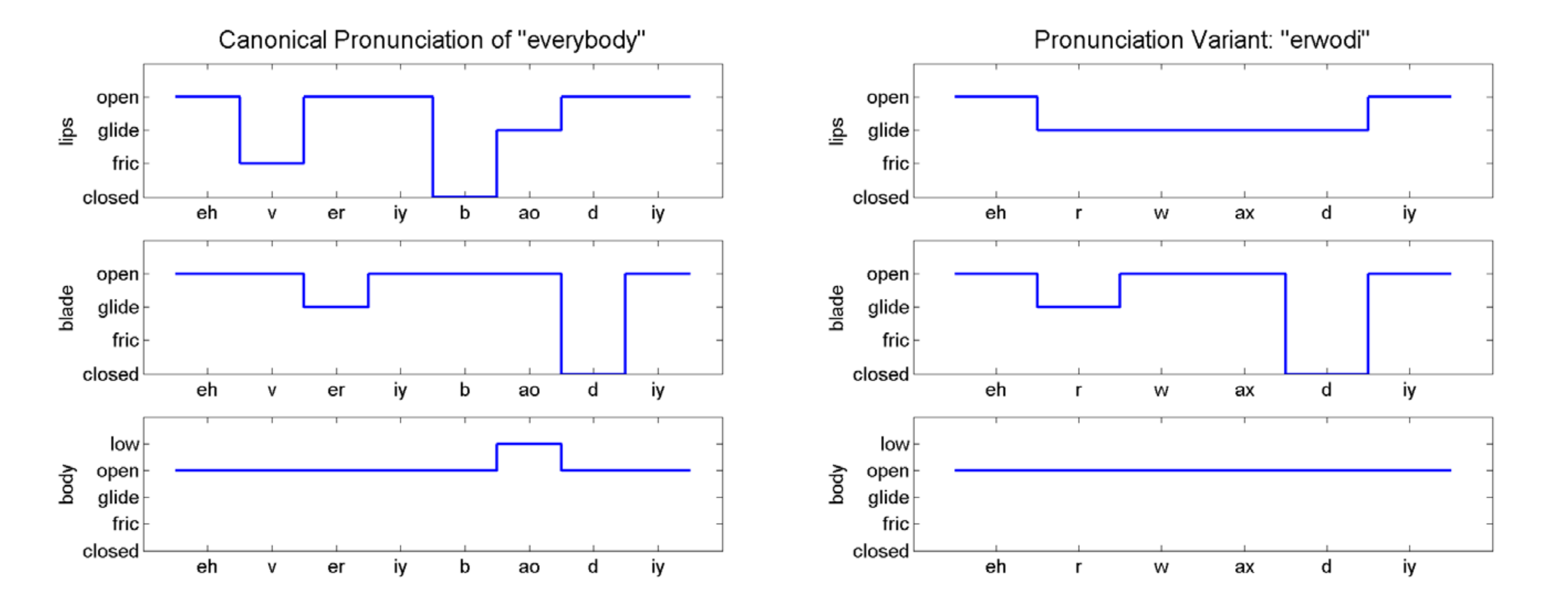


Event-Based Dynamic Programming smoothing of SVM outputs

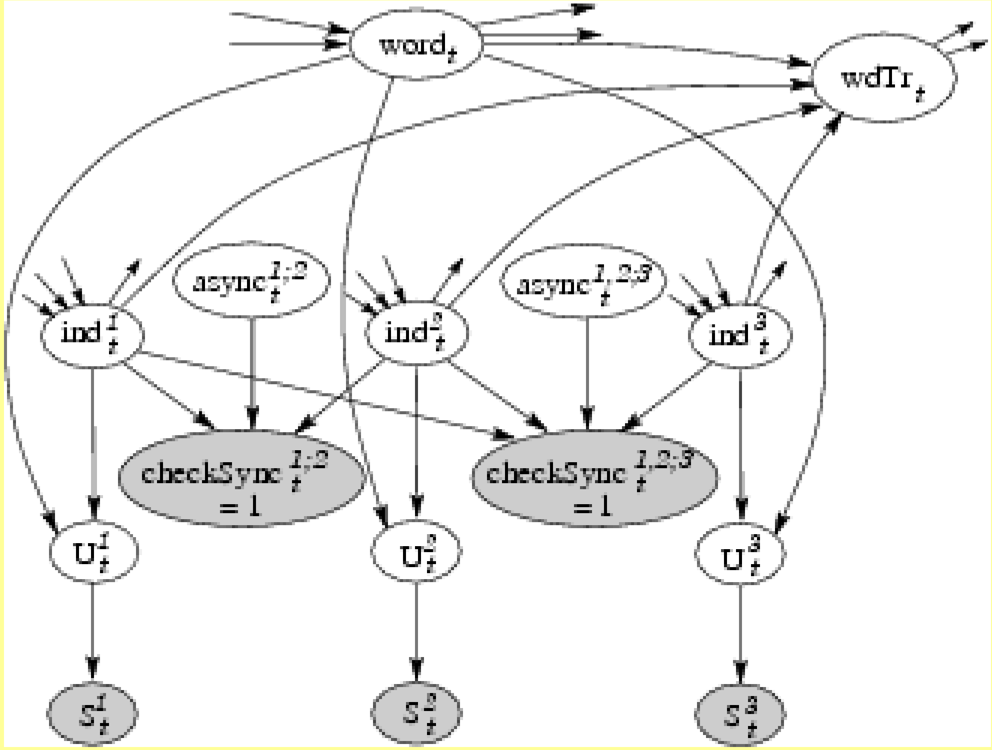
- Maximize $\Pi_i \; p(\text{ features}(t_i) \mid X(t_i)\,) \; p(t_{i+1}-t_i \mid \text{features}(t_i))$
- Forced alignment mode:
  computes p(word | acoustics); rescores the word lattice
- Manner class recognition mode:
  smooths SVM output; preprocessor for the DBN



Pronunciation Model: Dynamic Bayesian Network, with Partially Asynchronous Articulators



Pronunciation Model: DBN, with Partially Asynchronous Articulators



- $word_t$: word ID at frame #t
- $wdTr_t$: word transition?
- ind_t^i: which gesture, from the canonical word model, should articulator i be trying to implement?
- $async_t^{i;j}$: how asynchronous are articulators i and j?
- U_t^i: canonical setting of articulator #i
- S_t^i: surface setting of articulator #i

2. Experiments that have been used in rescoring

A. SVM training: Switchboard vs. NTIMIT
B. Acoustic features: MFCC vs. rate-scale
C. Training the pronunciation model
D. Event-based smoothing with and without pronunciation model
E. WER Reductions so far: summary

SVM Training: Switchboard vs. NTIMIT, Linear vs. RBF

- NTIMIT:
  - Read speech = reasonably careful articulations
  - Telephone-band, with electronic line noise
  - Transcription: phonemic + a few allophones
- Switchboard:
  - Conversational speech = very sloppy articulations
  - Telephone-band, electronic and acoustic noise
  - Transcription: reduced to TIMIT-equivalent for this experiment, but richer transcription available

SVM Training: Accuracy, per frame, in percent

Train	NTIMIT		NTIMIT&SWB		NTIMIT		Switchboard	
Test	NTIMIT		NTIMIT&SWB		Switchboard		Switchboard	
Kernel	Linear	RBF	Linear	RBF	Linear	RBF	Linear	RBF
speech onset	95.1	96.2	86.9	**89.9**	71.4	62.2	81.6	81.6
speech offset	79.6	88.5	76.3	**86.4**	65.3	78.6	68.4	83.7
consonant onset	94.5	95.5	91.4	93.5	70.3	72.7	95.8	**97.7**
consonant offset	91.7	93.7	94.3	**96.8**	80.3	86.2	92.8	**96.8**
continuant onset	89.4	94.1	87.3	**95.0**	69.1	81.9	86.2	92.0
continuant offset	90.8	94.9	90.4	**94.6**	69.3	68.8	89.6	94.3
sonorant onset	95.6	97.2	**97.8**	96.7	85.2	86.5	96.3	96.3
sonorant offset	95.3	96.4	94.0	**97.4**	75.6	75.2	95.2	96.4
syllabic onset	90.7	95.2	91.4	**95.5**	69.5	78.9	87.9	92.6
syllabic offset	90.1	88.9	87.1	**92.9**	54.4	60.8	88.2	89.7

Acoustic Feature Selection: MFCCs, Formants, Rate-Scale

1. Accuracy per Frame, Stop Releases only, NTIMIT

	MFCCs+Shape		MFCCs+Formants	
Kernel	Linear	RBF	Linear	RBF
+/- lips	78.3	90.7	92.7	**95.0**
+/- blade	73.4	**87.1**	79.6	85.1
+/- body	73.0	85.2	85.7	**87.2**

2. Word Error Rate: Lattice Rescoring, RT03-devel, One Talker
(WARNING: this talker is atypical.)

Baseline: 15.0% (113/755)

Rescoring, place based on:

MFCCs + Formant-based params: 14.6% (110/755)

Rate-Scale + Formant-based params: 14.3% (108/755)

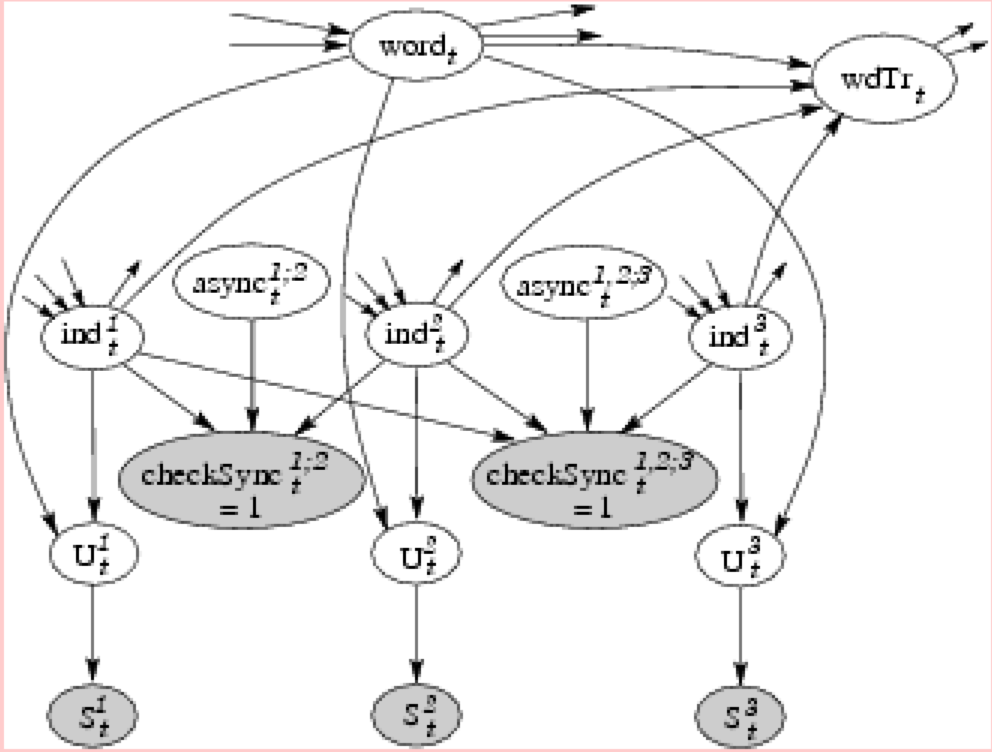
Event-Based Smoothing of SVM outputs with and without pronunciation model

1. No event-based smoothing
   - Manner-class recognition results: very bad (many insertions)
   - Lattice rescoring results: not computed

2. Event-based smoothing with no pronunciation model (no DBN)
   - Computational complexity: 30 seconds/lattice, 24 hours/RT03

3. Event-based smoothing followed by pronunciation model (DBN):
   - Computational complexity: 40 mins/lattice, 2000 hours/RT03

Training the Pronunciation Model



- Trainable Parameters:
  - **$p(ind^i_t|ind^i_{t-1})$**
  - **$p(U^i_t|ind^i_t,word_t)$**
  - **$p(async_t^{i,j}=d)$**
  - **$p(S^i_t|U^i_t)$**

- Experiment:
  - Train p(async) using manual transcriptions of Switchboard data
  - Test in rescoring pass, RT03, with SVM outputs

WER Results so far

	WER – 1 talker	WER – 27 talkers	Improved Talkers	Unchanged Talkers
Baseline	15.0%	20.3%	-	-
Rescored	14.6	-	-	-
Rate-scale+ Formant-based	14.3	-	-	-
DBN Trained	13.9	20.4%	6/27	12/27

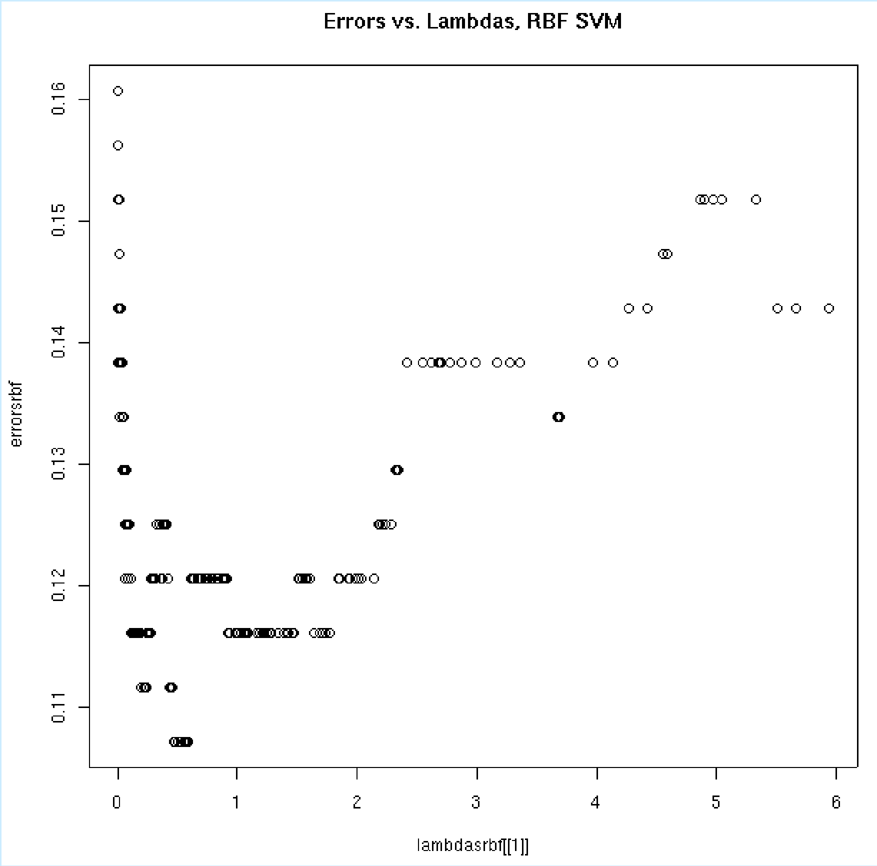
3. Ongoing Experiments: Acoustic Modeling

A. Acoustic feature vector size
B. Optimal regularization parameter for SVMs
C. Function words
D. Detection of phrasal stress

Acoustic Feature Vector Size: Accuracy/Frame, linear SVM, trained w/3000 tokens

Observation Vector Dimension	539 mfcc+formants	2000 …+shape+APs	10000 …+rate-scale
speech onset	86.9	**93.0**	77.6
speech offset	76.3	**95.3**	79.4
consonant onset	**91.4**	89.7	86.3
consonant offset	**94.3**	81.1	78.8
continuant onset	**87.3**	84.7	73.9
continuant offset	90.4	**91.5**	82.3
sonorant onset	**97.8**	83.8	81.1
sonorant offset	**94.0**	92.4	87.2
syllabic onset	**91.4**	85.2	73.8
syllabic offset	87.1	**88.0**	76.8

Optimal Regularization Parameter for the SVM



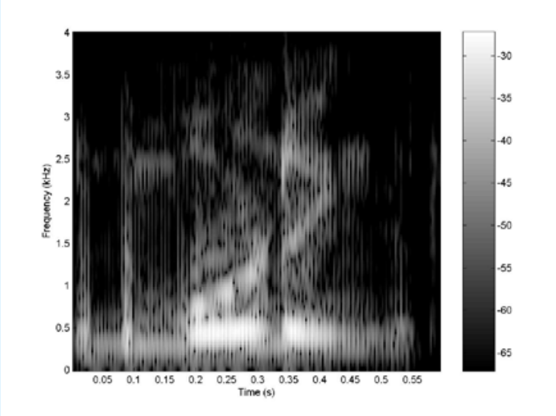
- SVM minimizes Train_Error+l*Generality
- If you trust your training data, choose a small l
- Should you trust your training data? Answers:
  1. OLD METHOD: Exhaustive testing of all possible ls
  2. NEW METHOD (Hastie et al.) simultaneously computes SVMs for all possible ls

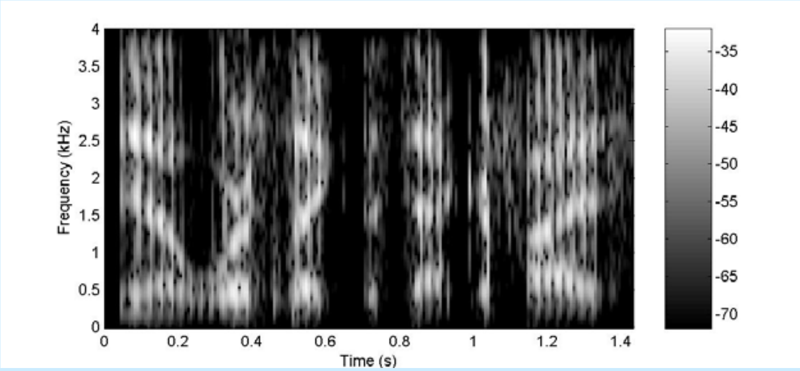
Analysis and Modeling of Function Words

- Function words account for most substitution errors in the SRI lattices:
  - it→that,99 (1.78%); the→a,68 (1.22%); a→the,68 (1.03%)
  - and→in,64 (1.15%); that→the,40 (0.72%); the→that,35 (0.63%)
- Possible Solutions
  - Model multiwords in the DBN, e.g. "IN_THE ih n dh ax" - DONE
  - Define SVM context to depend on function vs. content word – NOT YET
  - Better models of "partially deleted" phonemes, e.g. /dh/ *(that ↔ it, the ↔ a)*, /n/ *(you know → yõw)*

Better Models of "Partially Deleted" Phonemes

- Example: /dh/ is frequently a nasal (*in the*) or a stop (*at the*), but always implemented with a dental place of articulation (Manuel, 1994)
  - Conclusion: existence of "the" is cued by dental place of articulation of any consonant release
  - DBN could model manner change if given training data, but NTIMIT notation quantizes all /dh/ as either /dh/, /d/, or /n/
  - Possible solution: train [+dental] as a feature of all [+blade] consonants, regardless of manner – training tokens are all "fricative," but test tokens may be nasal or stop. DBN recognizes that manner of /dh/ is variable…
- Example: /n/ is deleted in "you know" or "I know," but leaves behind a nasalized vowel. Possible solution: recognize nasality of the vowel; DBN can attribute nasality of the vowel to a deleted nasal consonant.





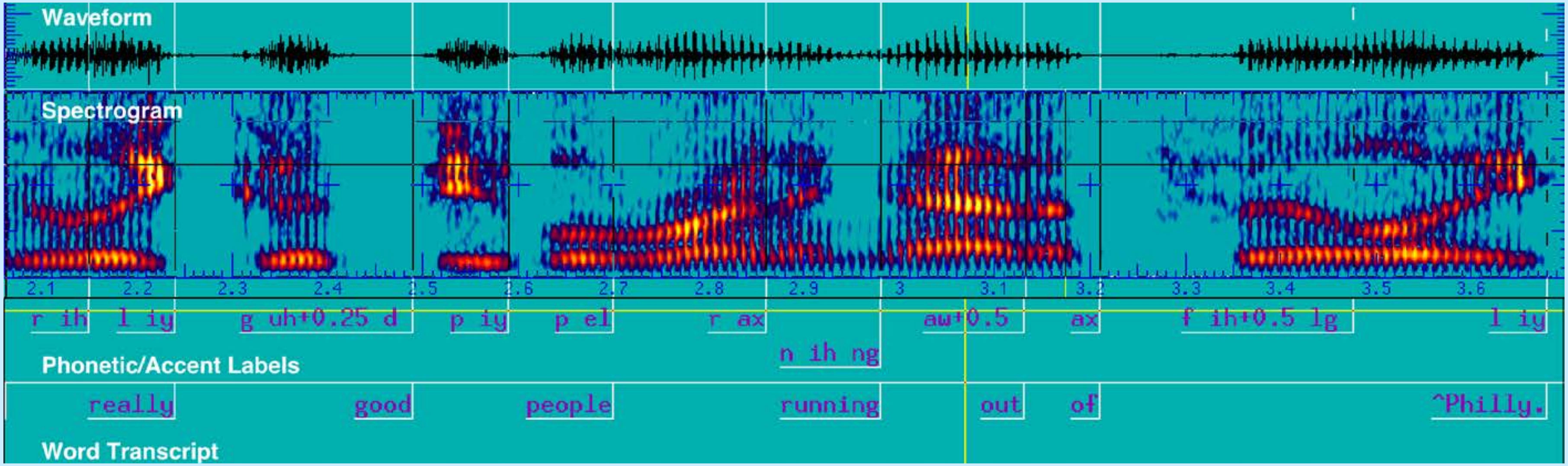
Detection of Phrasal Stress

The probability of a deletion error is MUCH higher in unstressed syllables

SVM detectors for phrasal stress (based on ICSI transcribed data) are currently under development

Phrasal stress distinguishes words: some syllable nuclei are allowed to carry phrasal stress, some are not

Phrasal stress conditions other pronunciation probabilities: it can identify words subject to increased probability of phoneme deletion.



4. Ongoing Experiments: Pronunciation Modeling

- Complexity Issues:
  - Improved triangulation of the DBN
  - Which reductions should we model?
- Discriminative Pronunciation Modeling:
  - A distinctive feature lexicon, with features added discriminatively to improve system performance
  - Discriminative optimization of pronunciation string probabilities using maximum entropy, conditional random fields
  - Discriminative models of landmark insertion, substitution, and deletion: a factored N-gram language model

Improved Triangulation of the DBN

- The DBN Inference Algorithm: $p(word_t \mid observations)$ is computed using the following algorithm:
  1. Triangulate so that cliques can be eliminated one at a time
  2. Marginalize over the cliques, one at a time, starting with the cliques farthest from $word_t$, until the only remaining variable is $word_t$
- Complexity of inference α $|S|^{NumVarPerClique}$
- Different triangulations result in different NumVarPerClique
- Finding the perfect triangulation is NP-hard
- Finding an OK triangulation:
  1. Start with initial guess about where the borders are between groups of variables
  2. Specify the flexibility of each border
  3. Search within specified limits
- Status: job is running (currently on day 7)

Which Reductions Should we Model?

- Virtually anything can reduce in natural speech due to stylistic, lexical, and phonological factors (Raymond et al. 2003). The problem: Every degree of freedom in $p(S^i_t|U^i_t)$ increases complexity of the DBN. Which of the possible reductions are most important?
- Common environments for reduction: (Greenberg et al. 2002; 2003)
  - Unstressed syllables
  - Syllable codas
- Segment types more prone to reduction:
  - Coronals: /t/, /d/, /n/, /s/
- Types of reductions commonly observed:
  - Absolute reduction = deletion
  - Other reductions: flapping, frication, etc.
- Based on these observations, we should model reduction and deletion of coda coronals (and related effects on preceding vowel formants), especially in unstressed syllables

Discriminative Pronunciation Modeling

We only need to distinguish between small sets of confusable words during rescoring, so … find a model that emphasizes landmark features relevant for distinguishing between words, train discriminatively.

1. Lexical representation:
   - ⇒ Select distinctive features that maximally discriminate confusable words
2. Computing p(pronunciation | word) discriminatively:
   - ⇒ (a) convert each word to a fixed-length landmark-based representation and use discriminative classifier (maxent)
   - ⇒ (b) use a discriminative sequence model (conditional random field)
   - ⇒ (c) represent the landmarks as “words” in a language model; apply discriminative language modeling techniques

Discriminative Selection of Distinctive Features

- A distinctive feature lexicon already exists, based on the Juneja-Espy feature set.
- Goal: add partially redundant binary features to each phoneme, in order to increase the likelihood of accurate lexical matches.
  – Discriminative selection using MAXENT (next slide)
  – Selection based on Switchboard error analysis, e.g. length, energy contour,

“Today”

Syllable	Manner	Place	Height	Glide	Voicing	Length	Energy	Accent	Segment
ON	ST	Central	*	–	–	#	#	–	t
NU	VO	Central	Hi	–	+	#	#	–	ax
ON	ST	Central	*	–	+	#	#	+	d
NU	VO	Front	Mid	+	+	#	#	+	eh
NU	VO	Front	Hi	+	+	#	#	+	ih

“Ready”

Syllable	Manner	Place	Height	Glide	Voicing	Length	Energy	Accent	Segment
ON	RH	back	*	+	+	#	#	+	r
NU	VO	front	Hi	–	+	#	#	+	ax
JU	FLAP	*	*	–	*	#	#	*	dx
NU	VO	front	Hi	+	+	#	#	–	ih
NU	VO	front	Hi	+	+	#	#	–	iy

Discriminative Optimization of Pronunciation Probabilities Using Maximum Entropy

- Convert word lattices to confusion networks (SRI-style)
- For each confusion set, train maxent model on landmark representation:

$$p(y \mid x) = \frac{1}{Z(x)} \exp\left(\sum_{i=1}^{k} \lambda_i f_i(x, y)\right)$$

- y: word, x: landmark sequence, f(y,x): function indicating presence/absence/frequency of basic temporal relation (precedence, overlap) between two landmarks
- Apply model to landmark detector output
- Interpolate resulting probabilities with posterior word probabilities from confusion network and rescore

Discriminative Optimization of Pronunciation Probabilities Using Conditional Random Fields

- Use graph structure similar to that in DBN, with one primary landmark stream defining state sequence
- Other landmarks are treated as feature functions
- Train using CRFs:

$$p(y \mid x) = \frac{1}{Z(x)} \exp\left(\sum_t \sum_k \lambda_k f_k(y_{(k,t)}, x, t)\right)$$

- y: word state sequence, x: landmark sequence, t: length, k: feature dimensionality
- add scores to lattices or n-best lists and rescore

Landmark N-gram Pronunciation Model

WORD completely 20050 20710

MANNER +-continuant -+continuant:+voice +syllabic -+sonorant:+voice +-sonorant:-voice +syllabic -+sonorant:-voice +-sonorant:-voice -+sonorant:-voice +-continuant -+continuant +syllabic -+sonorant -+sonorant:-voice +syllabic -+continuant +-sonorant:+voice +syllabic +-continuant +syllabic +-continuant -+continuant -+continuant -+sonorant:-voice +syllabic +syllabic

PLACE +lips +lips +front:-high -strident:+anterior +strident:+anterior -front:+high +strident:+anterior +strident:-anterior -strident:+anterior +lips +body -front:-high +strident:+anterior -front:+high -nasal:+blade -strident:+anterior +front:+high -nasal:+blade -front:+high +lips +lips -nasal:+blade +strident:+anterior +front:-high +front:+high

- *Main idea: Model sequences of landmarks for words and phones*
- *Approach: Train word and phone landmark N-gram LMs to generate a smoothed backoff LM*
  - *For common words, train word landmark LMs*
  - *For context dependent phones, train CDP landmark LMs*
  - *For all monophones, train phone landmark LM's*
  - *Score each word in a smoothed manner with word, CDP, and phone LMs*

5. Ongoing Experiments: Rescoring Methods

1. Recognizer-generated N-best sentences vs. Lattice-generated N-best sentences
2. Maximum-entropy estimation of stream weights

Lattices and N-best Lists

- Basic Rescoring Method:

  word_score = a*AM + b*LM + c*#words+ d*secondpass
- Estimation of stream weights is correctly normalized for N-best lists, not lattices
- Two methods for generating N-best:
  - Run recognizer in N-best mode
  - Generate from lattices

	N-best from Recognizer	N-best from Lattices
WER based on 1st-pass recognizer scores	24.4%	24.1%

Maximum Entropy Estimation of Stream Weights

- Conditional exponential model of score combination estimated by Maximum Entropy[1]
- Set of feature functions:

$$f_1(obs, hyp) = \log p_{AM}(obs \mid hyp)$$

$$f_2(obs, hyp) = \log p_{LM}(hyp)$$

$$f_3(obs, hyp) = [\# words(hyp)]$$

$$f_4(obs, hyp) = \log p_{LANDMARK-PRONUNCIATION-MODEL}(obs \mid hyp)$$

$$\log P(hyp \mid obs) = \left(\sum_i \lambda_i f_i(obs, hyp)) \right) - \log Z(obs)$$

[1] Yu, Waibel ICASSP 2004

Maximum Entropy Estimation of Stream Weights

- Computation of the partition function (normalization factor)

$$Z(obs) = \sum_{hyp(N-best)} \exp\left(\sum_i \lambda_i f_i(obs, hyp)) \right)$$

- Tool: MaxEnt program by Zhang Le
  - Optimization by L-BFGS algorithm for continuous variables
- Currently, experimenting with various normalizations of the scores
  - Positive, normalized features, appropriate definition of labels and proper approximation of the partition function necessary
  - Experiments continuing

Conclusions (so far)

- WER reduced for the lattices of one talker
- Computational complexity inhibits full-corpus rescoring experiments
- Ideas that may help reduce WER:

1. Discriminative pronunciation modeling
2. Discriminative combination of pronunciation models
3. Fine phonetic distinction
    - The right acoustic features for the right job
    - Detect distinctive features that have been “cut free” from a deleted segment, e.g., [+dental] of /dh/ in “in the,” or [+nasal] of /n/ in “you know.” Pronunciation model should use these “cut free” distinctive features to cue existence of a deleted phone
4. Teach people to enunciate more clearly