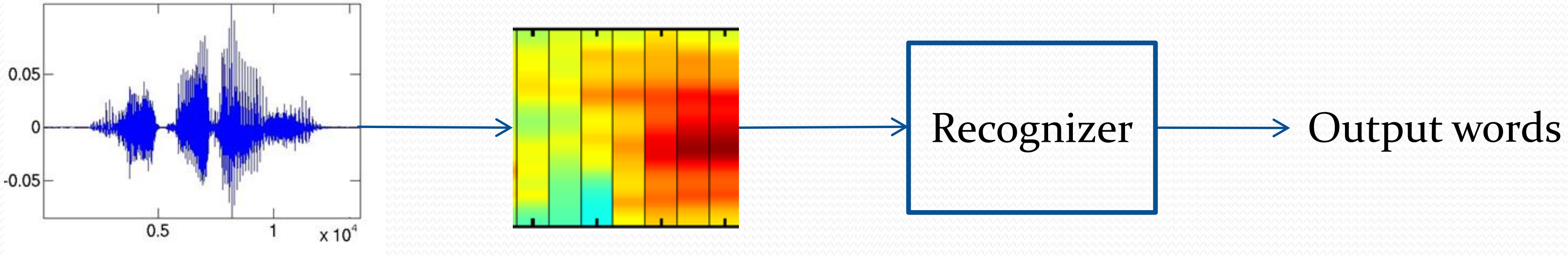
# Speech Recognition with Segmental Conditional Random Fields

# The Team !

- Senior Members
  - Les Atlas, University of Washington
  - Kris Demuynck, Leuven University
  - Hynek Hermansky, JHU
  - Aren Jansen, JHU COE
  - Damianos Karakos, JHU
  - **Patrick Nguyen**, Microsoft Research
  - Fei Sha, USC
  - Dirk Van Compernolle, Leuven
  - **Geoffrey Zweig**, Microsoft Research
- Student Members
  - Sam Bowman, University of Chicago
  - Pascal Clark, UW
  - Sivaram GSVS, JHU
  - Justine Kao, Stanford
  - Greg Sell, Stanford
  - Samuel Thomas, JHU
  - Meihong Wang, USC
- Thanks!
  - Brian Kingsbury
  - IBM Research
  - Ken Church

# The Problem

- State-of-the-art speech recognizers look at speech in just one way
  - Frame-by-frame
  - With one kind of feature



Recognizer

Output words

- And often the output is wrong

**“Oh but he has a big challenge”**

**What we want (what was said)**

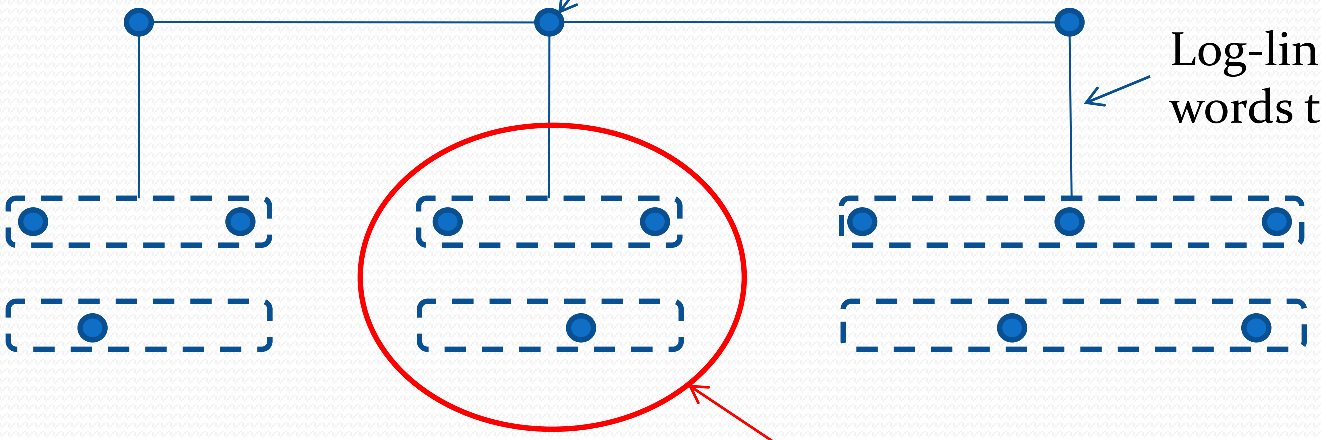
≠

**“ALREADY AS a big challenge”**

**What we get**

# The Goal

- Look at speech in multiple ways
- Extract information from multiple sources
- Integrate them in a segmental, log-linear model



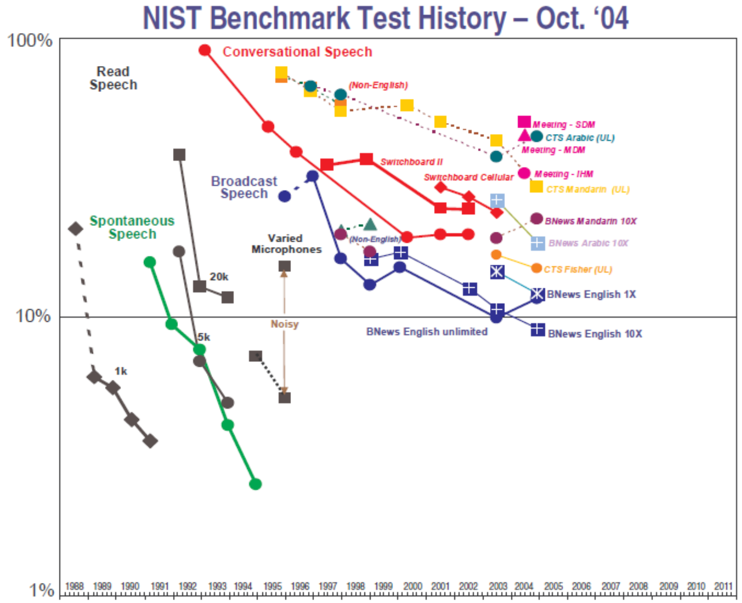
States represent whole words (not phonemes)
Baseline system can constrain possibilities

Log-linear model relates
words to observations

**Multiple information sources,
e.g. phoneme, syllable detections**

# Data Sets

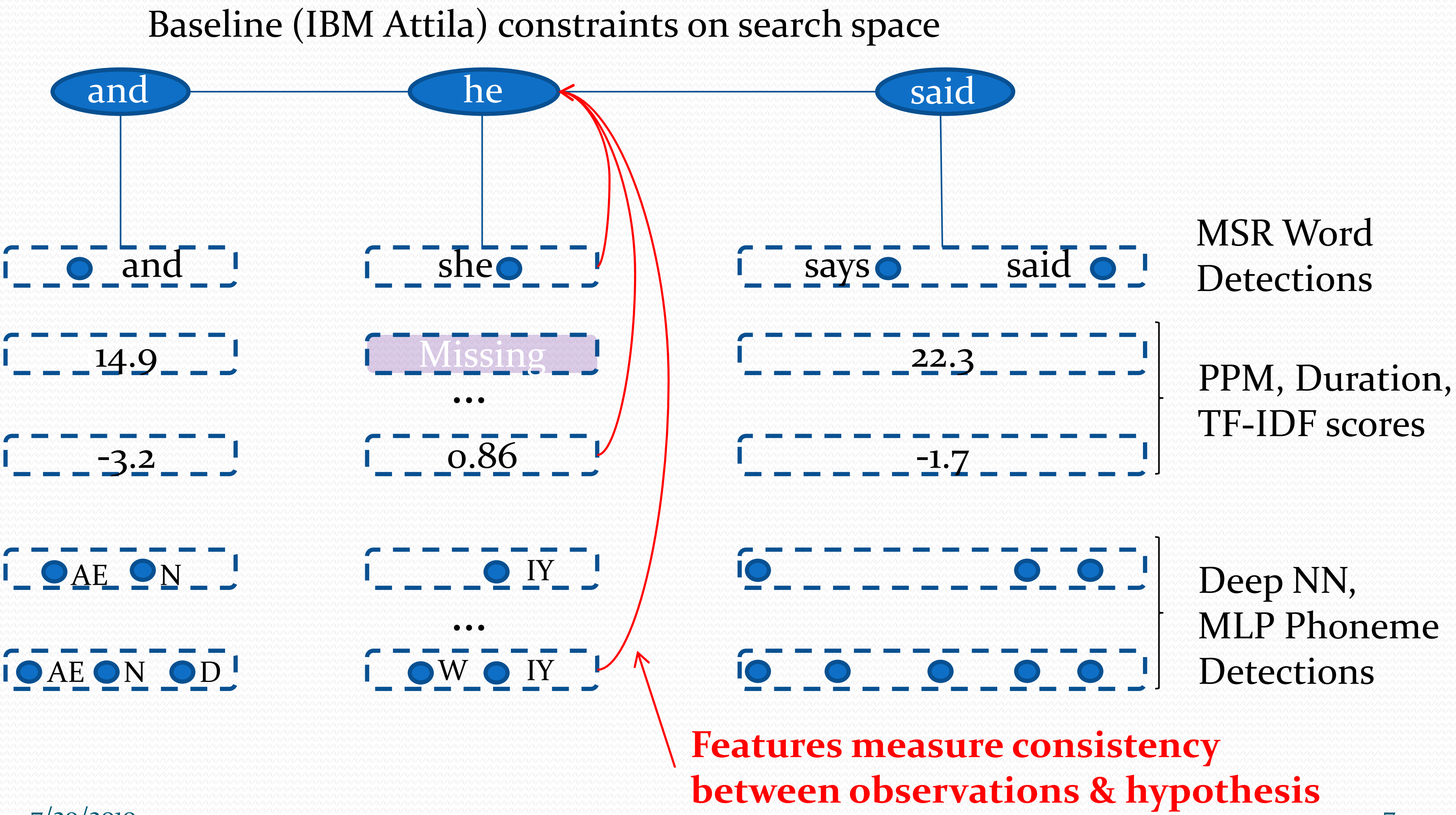
- Wall Street Journal
  - Read newspaper articles
  - 81 hrs. training data
  - 20k open vocabulary test set
- Broadcast News
  - 430 hours training data
  - ~80k vocabulary
- World class baselines for both
  - 7.3% error rate WSJ (Leuven University)
  - 16.3% error rate BN (IBM Attila system)



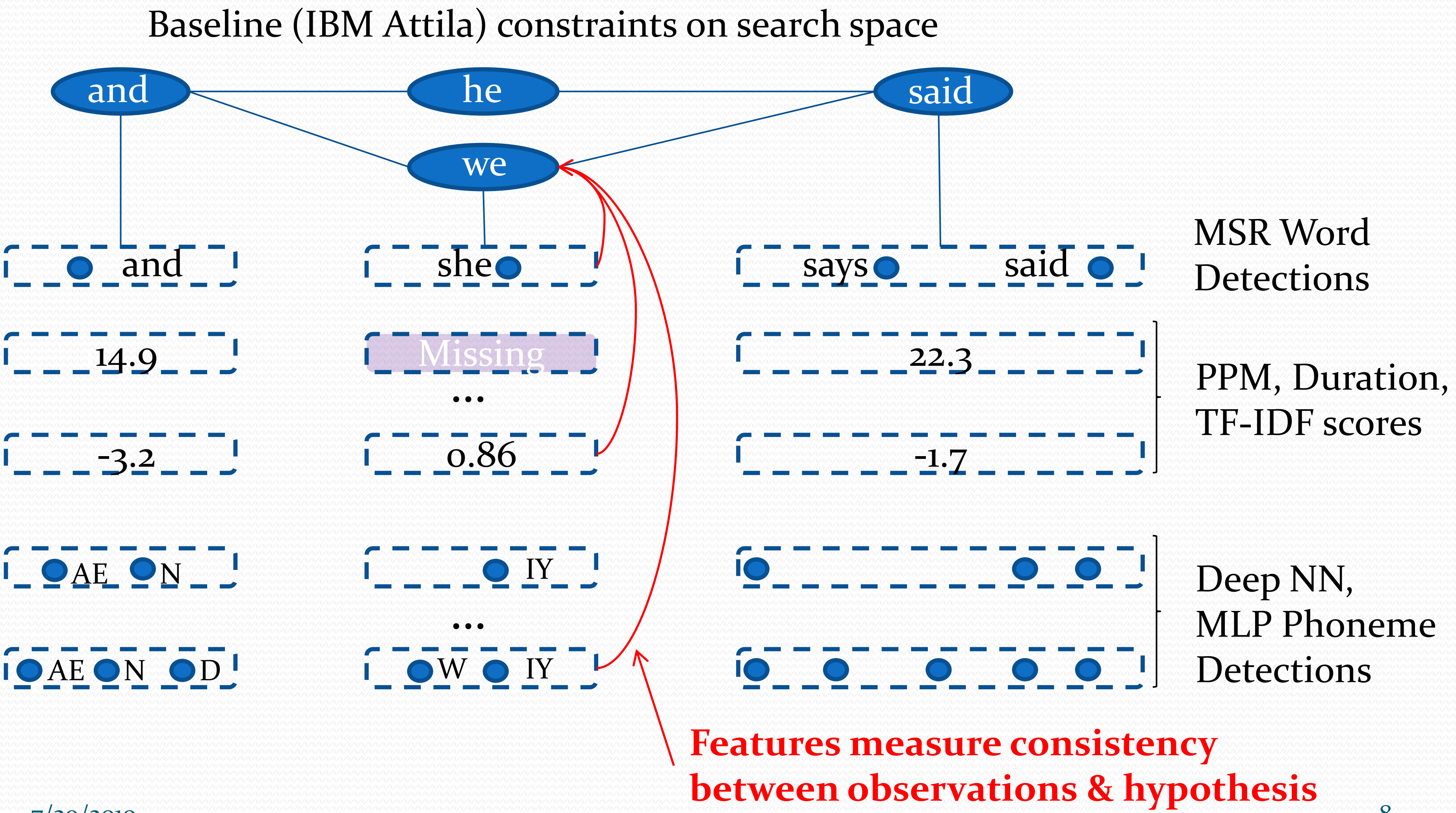
# Main Accomplishments (1)
# Integrating Framework for New Research

- Developed SCARF toolkit
- SCARF integrates
  - Multiple **types** of information
    - Binary event detections, e.g. phoneme detections
    - Real valued scores, e.g. Point Process Model scores
  - Information across **granularities**
    - Word, syllable, phoneme scales
  - Information of variable **completeness** and **quality**
    - Baseline: (~12% PER)
    - MSR Word detectors: (~15% PER)
    - Phoneme detectors: (~30% PER)
    - Point Process Model: (Partial annotation only)
- Difficult to do this conventionally
  - Segment level scores, correlated features

# Integrating Framework, High Level View


Baseline (IBM Attila) constraints on search space
and
he
said
and
she
says
said
MSR Word Detections
14.9
Missing
…
22.3
-3.2
0.86
-1.7
PPM, Duration, TF-IDF scores
AE
N
IY
…
AE
N
D
W
IY
Deep NN, MLP Phoneme Detections
Features measure consistency between observations & hypothesis

# Integrating Framework, High Level View


Baseline (IBM Attila) constraints on search space
and
he
said
we
and
she
says
said
MSR Word Detections
14.9
Missing
…
22.3
-3.2
0.86
-1.7
PPM, Duration, TF-IDF scores
AE
N
IY
…
AE
N
D
W
IY
Deep NN, MLP Phoneme Detections

**Features measure consistency between observations & hypothesis**

# Main Accomplishments (2)
# Improved on State-of-The-Art Baselines

| Wall Street Journal | WER | % Possible Gain |
|---|---|---|
| Baseline (SPRAAK / HMM) | 7.3% | 0% |
| + SCARF, template features | **6.7** | **14** |
| (Lattice Oracle – best achievable) | 2.9 | 100 |

| Broadcast News | WER | % Possible Gain |
|---|---|---|
| Baseline (Attila w/ VTLN, HLDA, fMLLR, fMMI, mMMI, MLLR) | 16.3% | 0% |
| + SCARF, word, phoneme detectors, scores | **15.0** | **25** |
| (Lattice Oracle – best achievable) | 11.2 | 100 |

# Main Accomplishments (2)
# Improved on State-of-The-Art Baselines

**Note improvement on top of discriminatively trained baseline !**

| **Broadcast News** | **WER** | **% Possible Gain** |
|---|---|---|
| Baseline (Attila w/ VTLN, HLDA, fMLLR, fMMI, mMMI, MLLR) | 16.3% ↓ | 0% |
| + SCARF, word, phoneme detectors, scores | **15.0** | **25** |
| (Lattice Oracle – best achievable) | 11.2 | 100 |

# Main Accomplishments (3) Advanced Cutting Edge Research

- Modulation Models of Speech
  - Compared the two most advanced approaches wrt LVCSR
  - Better scientific understanding of pitch-harmonic sampling
- Deep Neural Networks
  - From TIMIT to benefits in LVCSR
  - Developed architecture for running on standard CPU clusters
- MLP Posteriors
  - First use in LVCSR outside of Tandem NN+MFCC features
- Template Based Recognition
  - Showed benefits from spectrum of new features – e.g.
    - How many of the best matching examplars originated from the word to be recognized ?
- Point Process Phone Detectors
  - Showed benefit of word-level scores
  - Speedy, scalable implementation to scan large data sets

# Outline of Remainder

- SCARF Introduction (Patrick Nguyen) 10 min.
- Wall Street Journal / Template Results (Dirk Van Compernolle) 15 min.
- Broadcast News Fundamentals (Damianos Karakos) 5 min.
- Using Cohort Information (Damianos Karakos) 10 min.
- MLP Phoneme Detectors (Samuel Thomas) 15 min.
- Deep NN Phoneme Detectors (Fei Sha) 15 min.
- TF-IDF Acoustic Scores (Sam Bowman) 5 min.

Break

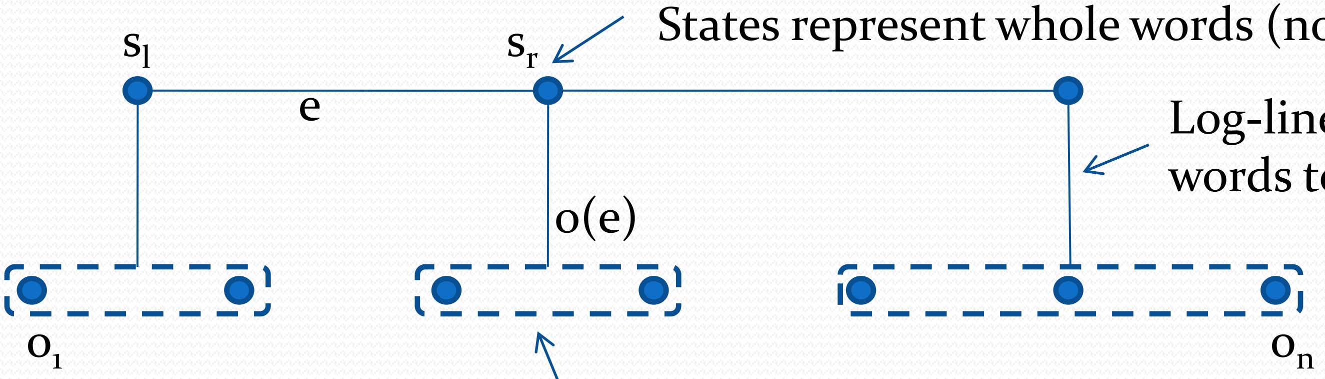
- Modulation Features (Pascal Clark) 15 min.
- Duration Models (Justine Kao) 10 min.
- Window-Based Detectors (Aren Jansen) 15 min.
- Summary (Geoffrey Zweig) 5 min.

# SCARF Introduction

Geoffrey Zweig

Patrick Nguyen

# Model Structure



States represent whole words (not phonemes)

Log-linear model relates words to observations

Observations blocked into groups corresponding to words. Observations typically detection events.

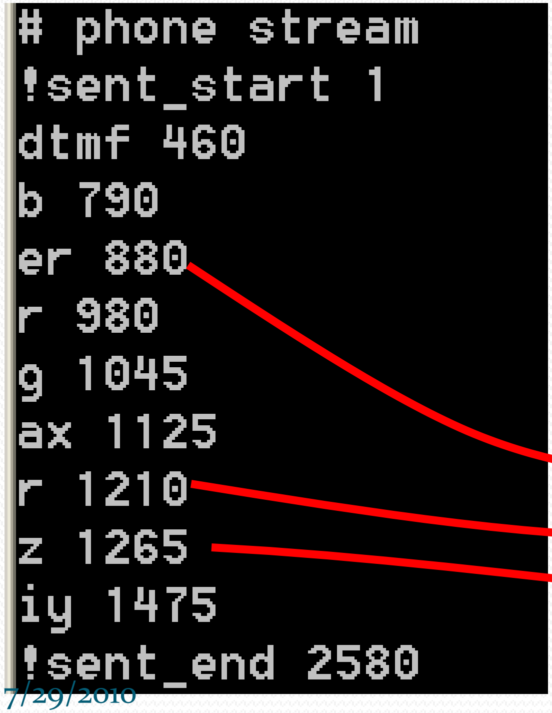
For a hypothesized word sequence s,
we must sum over all possible segmentations q of observations

$$P(\mathbf{s}|\mathbf{o}) = \frac{\sum_{\mathbf{q}\ s.t.\ |\mathbf{q}|=|\mathbf{s}|} \exp(\sum_{e\in\mathbf{q},k} \lambda_k f_k(s_l^e, s_r^e, o(e)))}{\sum_{\mathbf{s}'} \sum_{\mathbf{q}\ s.t.\ |\mathbf{q}|=|\mathbf{s}'|} \exp(\sum_{e\in\mathbf{q},k} \lambda_k f_k(s_l'^e, s_r'^e, o(e)))}$$

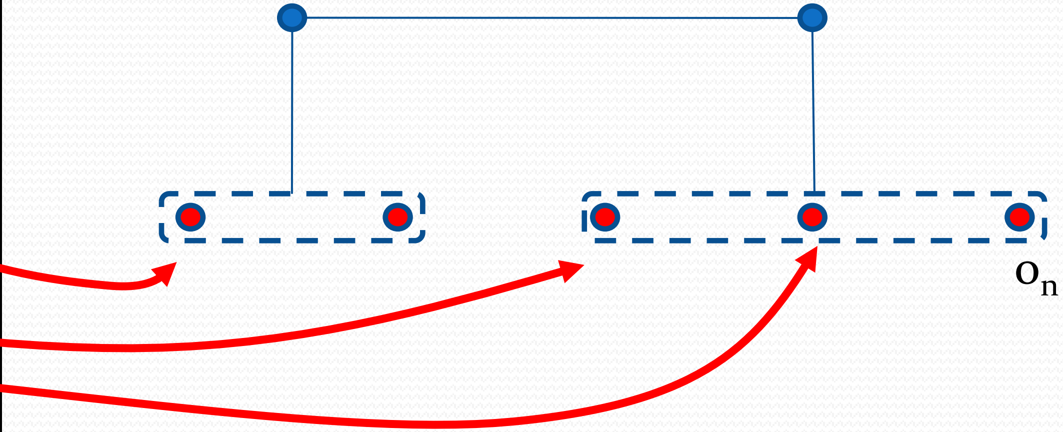
Training done to maximize product of label probabilities in the training data (CML).

# Inputs (1)

- Detector streams
  - (detection time) +
- Optional dictionaries
  - Specify the expected sequence of detections for a word

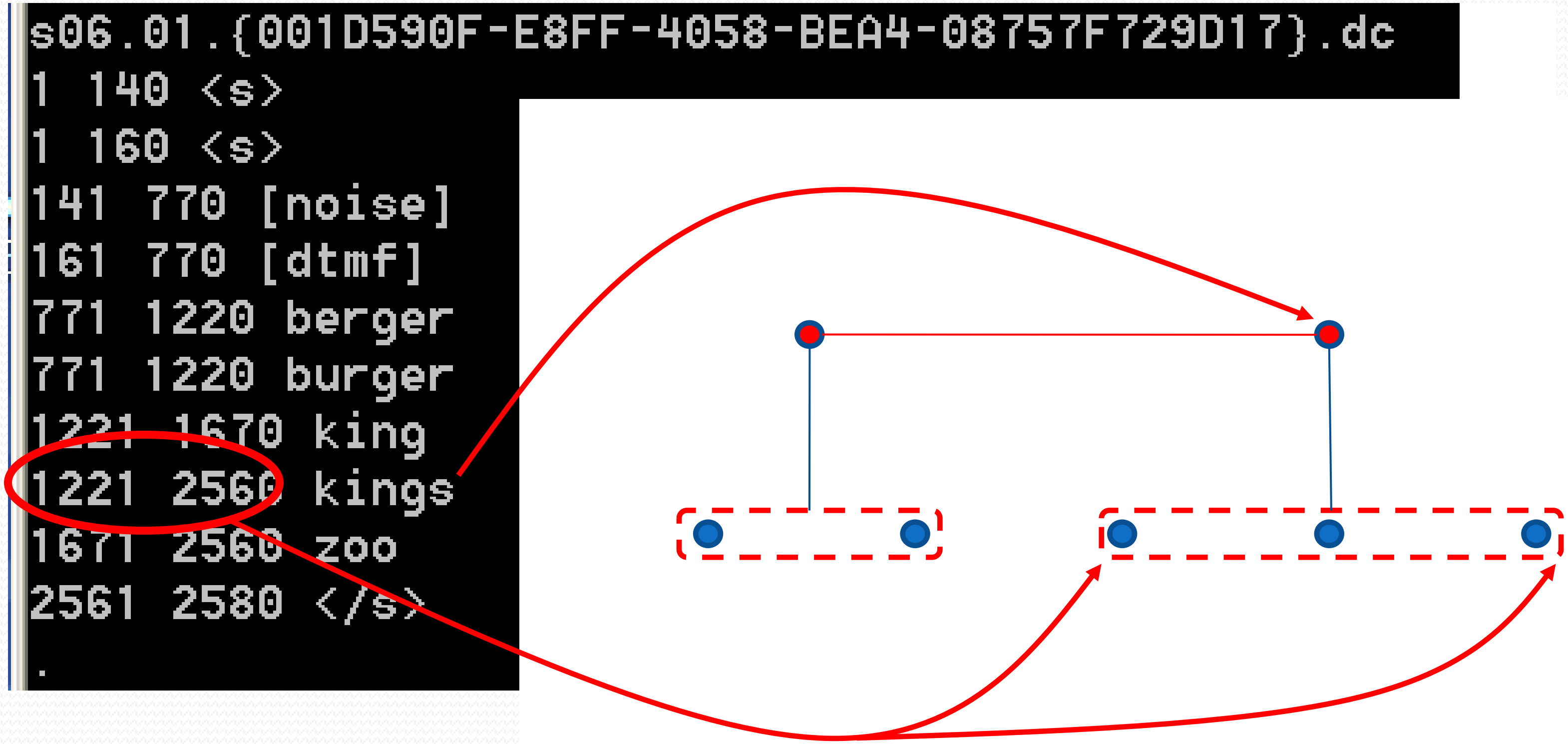


```
# phone stream
!sent_start 1
dtmf 460
b 790
er 880
r 980
g 1045
ax 1125
r 1210
z 1265
iy 1475
!sent_end 2580
```



# Inputs (2)

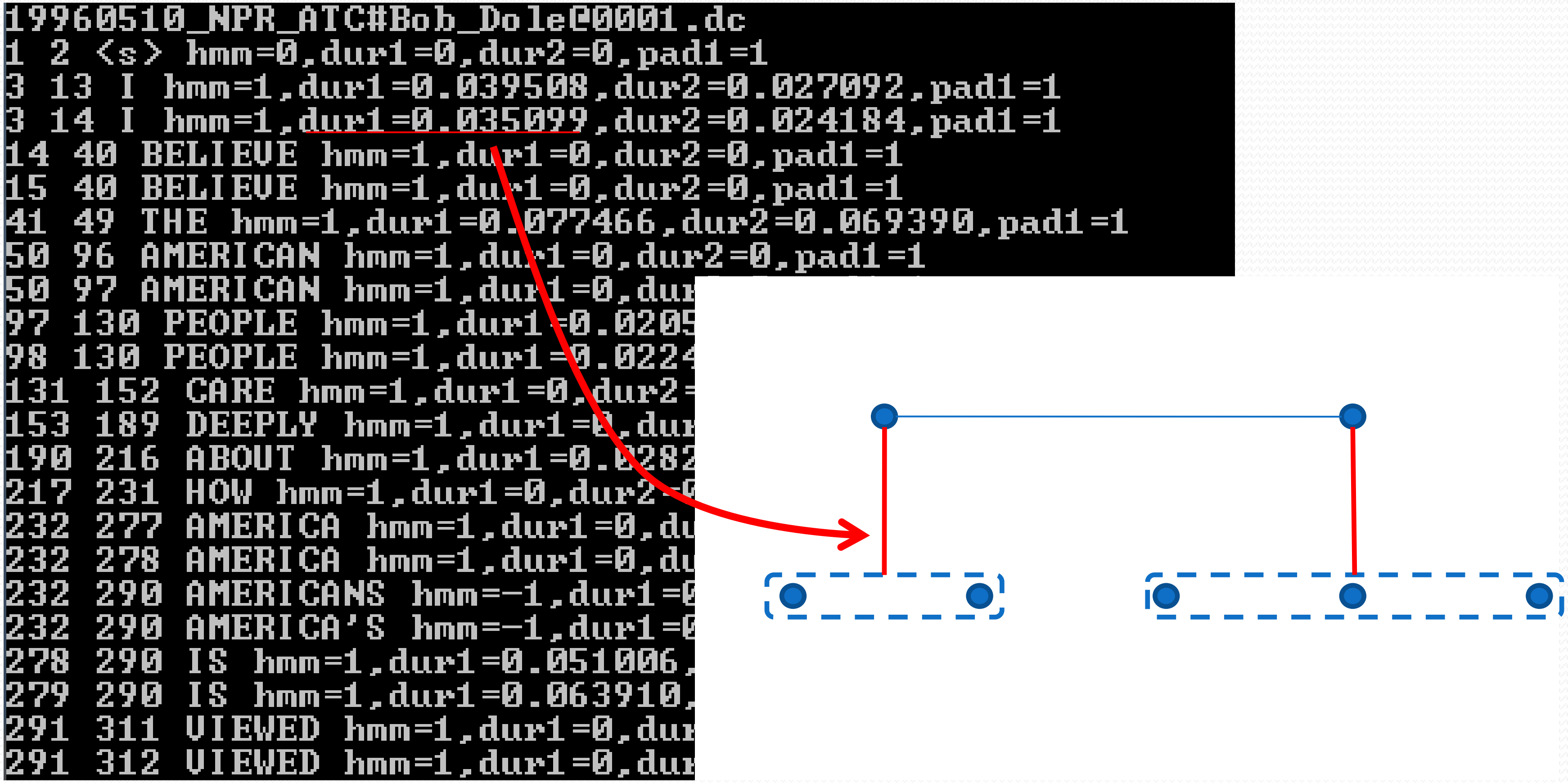
- Lattices to constrain search



```
s06.01.{001D590F-E8FF-4058-BEA4-08757F729D17}.dc
1  140  <s>
1  160  <s>
141  770  [noise]
161  770  [dtmf]
771  1220  berger
771  1220  burger
1221  1670  king
1221  2560  kings
1671  2560  zoo
2561  2580  </s>
.
```

# Inputs (3)

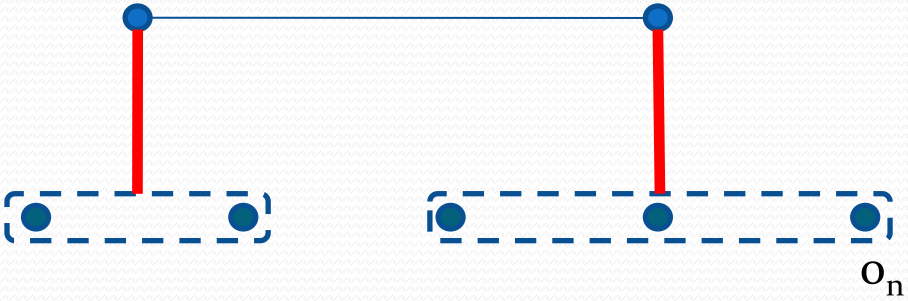
- User-defined features



```
19960510_NPR_ATC#Bob_Dole@0001.dc
1 2 <s> hmm=0,dur1=0,dur2=0,pad1=1
3 13 I hmm=1,dur1=0.039508,dur2=0.027092,pad1=1
3 14 I hmm=1,dur1=0.035099,dur2=0.024184,pad1=1
14 40 BELIEVE hmm=1,dur1=0,dur2=0,pad1=1
15 40 BELIEVE hmm=1,dur1=0,dur2=0,pad1=1
41 49 THE hmm=1,dur1=0.077466,dur2=0.069390,pad1=1
50 96 AMERICAN hmm=1,dur1=0,dur2=0,pad1=1
50 97 AMERICAN hmm=1,dur1=0,dur
97 130 PEOPLE hmm=1,dur1=0.0205
98 130 PEOPLE hmm=1,dur1=0.0224
131 152 CARE hmm=1,dur1=0,dur2=
153 189 DEEPLY hmm=1,dur1=0,dur
190 216 ABOUT hmm=1,dur1=0.0282
217 231 HOW hmm=1,dur1=0,dur2=
232 277 AMERICA hmm=1,dur1=0,du
232 278 AMERICA hmm=1,dur1=0,du
232 290 AMERICANS hmm=-1,dur1=
232 290 AMERICA'S hmm=-1,dur1=
278 290 IS hmm=1,dur1=0.051006,
279 290 IS hmm=1,dur1=0.063910,
291 311 VIEWED hmm=1,dur1=0,dur
291 312 VIEWED hmm=1,dur1=0,dur
```

# Detector-Based Features

- Array of features automatically constructed
- Measure forms of consistency between expected and observed detections
  - Differ in use of ordering information and generalization to unseen words
- Existence Features
- Expectation Features
- Levenshtein Features
- Baseline Feature



# Levenshtein Features

- Match of u
- Substitution of u
- Insertion of u
- Deletion of u

| | | | | |
|---|---|---|---|---|
| Expected: | ax | k | or | d |
| Detected: | ih | k | or | * |

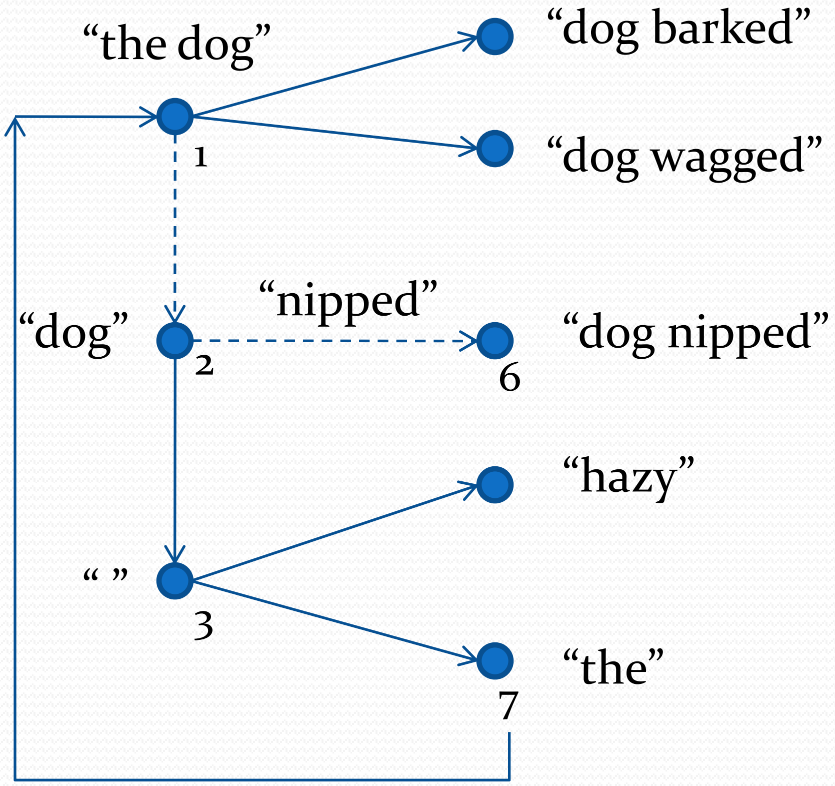
Sub-ax = 1
Match-k = 1
Match-or = 1
Del-d = 1

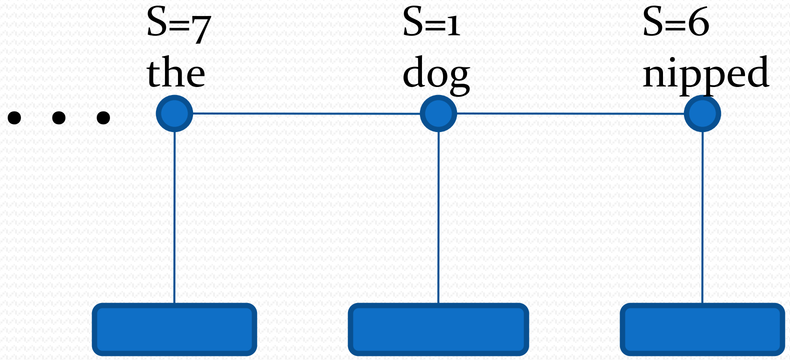
- Align the detector sequence in a hypothesized word’s span with the dictionary sequence that’s expected
- Count the number of each type of edits
- Operates only on the atomic units
- Generalization ability across words!

# The Baseline Feature

- The baseline feature treats the 1-best output of a baseline system as a detector stream
- The baseline feature is:
  - **+1 if a hypothesized word covers exactly one baseline detection, and words are the same**
  - **Otherwise it is -1**
- To maximize,
  - Hypothesis must have the same number of words as baseline,
  - And their identities must be the same
- With a high enough weight, the baseline output is guaranteed
- In practice, the weight is learned along with all the others

# Embedding a Language Model





At minimum, we can use the state sequence to look up LM scores from the finite state graph. These can be features.

And we also know the actual arc sequence. A 0/1 feature for each arc followed results in a discriminatively trained LM.

# Testing The Setup (1)

Can SCARF learn from correct detections?

| Setup | WER |
|---|---|
| Starting Point | 16.0% |
| + Oracle Detections | 11.8 |
| Lattice Oracle Error Rate | 11.2 |

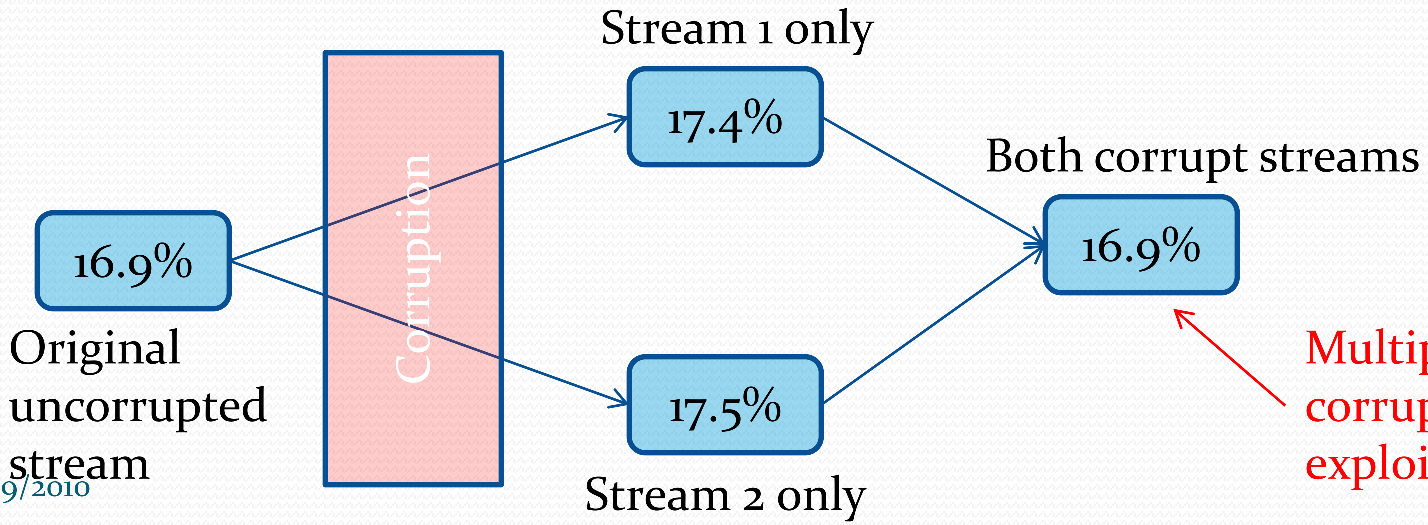
Yes - give it correct detections and you get correct words

(Modulo “break through” vs “breakthrough”, “Mohammed” vs “Muhammed”, etc.)

# Testing The Setup (2)

Can SCARF combine complementary information?

-Divide the phonemes into two sets
-Corrupt the baseline stream phonemes
-Detector stream 1 has all phonemes from set 1 corrupted
-Stream 2 has the others corrupted
-Train and decode with a unigram LM



Multiple uncorrelated corrupt streams exploited.

# Incorporating Template Based Features into the SCARF Framework

Kris Demuynck

Dirk Van Compernolle

Dino Seppi

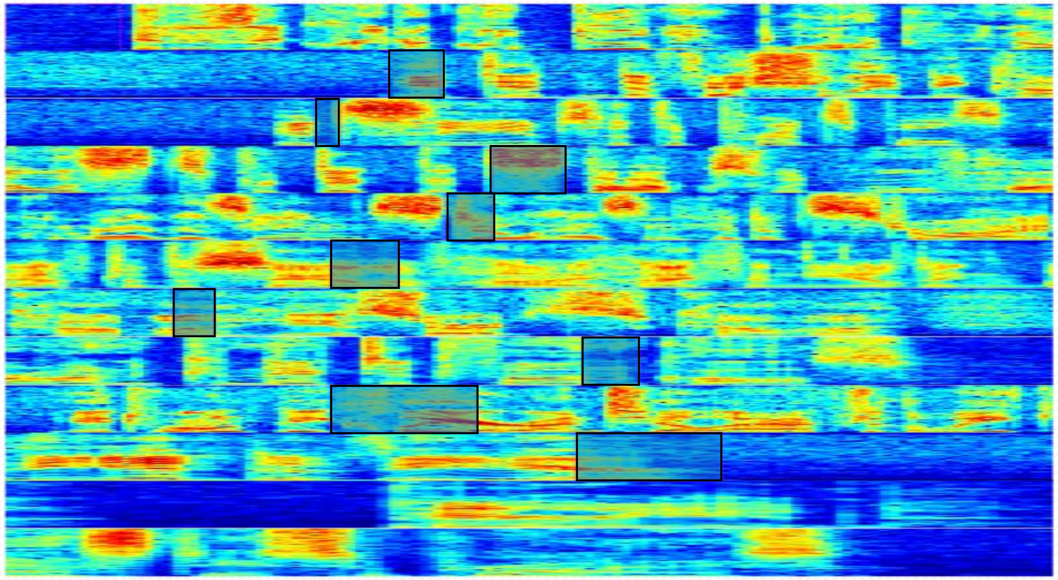
# Achievements

- basic improvements on our reference template based speech recognizer
- vast speedup of the template based system
- extracting & integrating multiple template based features  via the SCARF framework
- improve on the HMM baseline with added phone detectors via the SCARF framework
- combining HMM, DTW, KNN features via SCARF into a top performing system

# Template Based Recognition - Example

Speech Database
pre-segmented in templates
(phones)

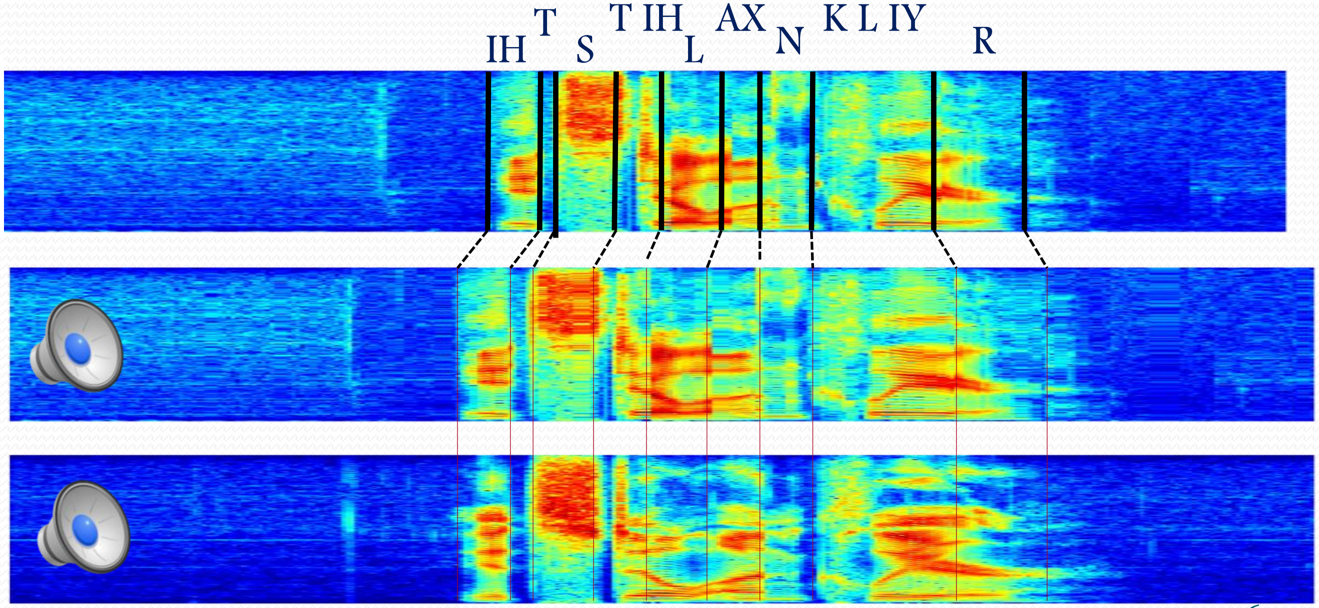
(12 x 2sec segments shown
of hrs of speech and
millions of templates )



Selected
Templates

Templates after
Dynamic Time Warping

Input Signal



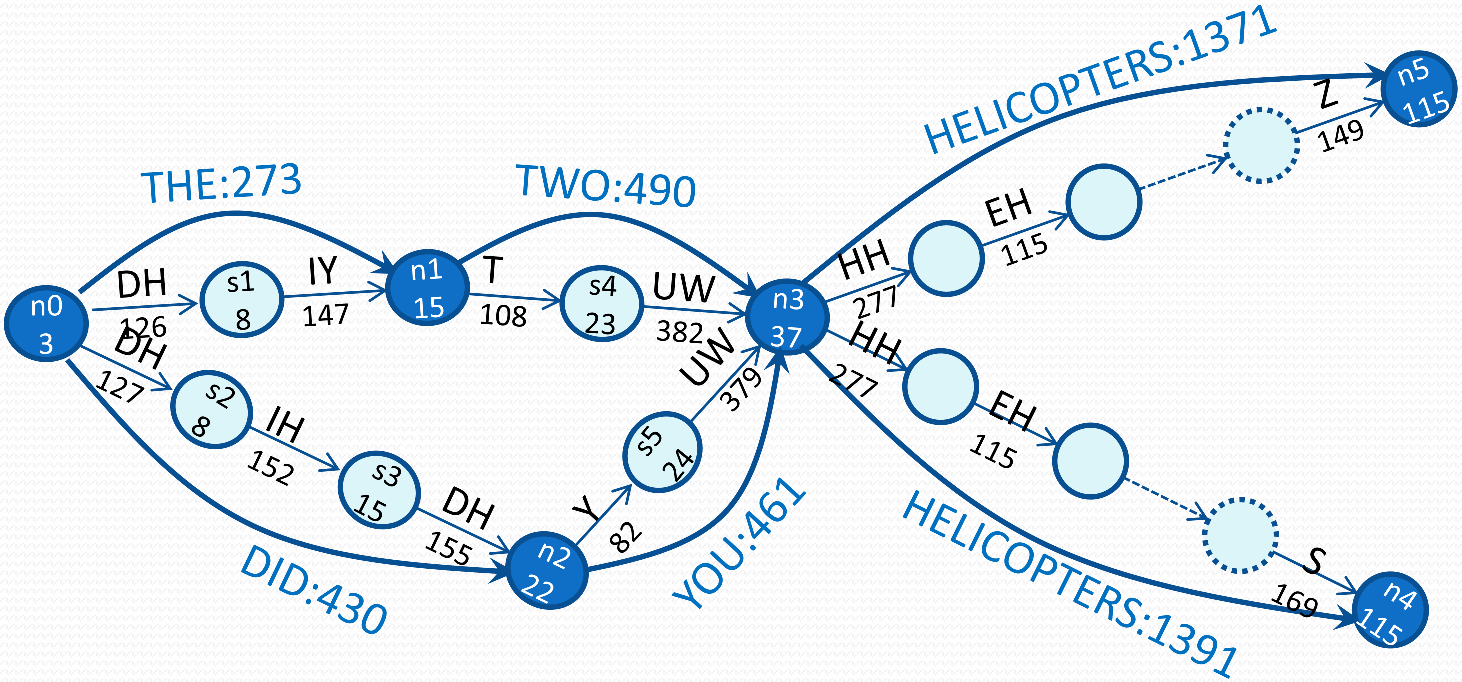
# Template Based Speech Recognition – Motivation & Concepts

- Motivation for a Template(=Example) Based Recognition:
  - doing away with the 1st order Markov assumption
  - exploit detail information available in the original data that gets blurred in the HMM density estimation
  - no assumption about the shape of the parametric densities

- SCARF:
  - WHY:
    - convenient framework to bring many diverse 'evidence streams' together
    - also breaks away from the 'sub-phonemic' HMM-state
  - HOW:
    - annotating the word lattices with novel parameterizations

- Challenges:
  - memory and CPU intensive
  - sensitivity to outliers
  - non-trivial integration of intermediate KNN info into single best decoding strategy

# How it Works



MAIN Structure: word graph with score annotation

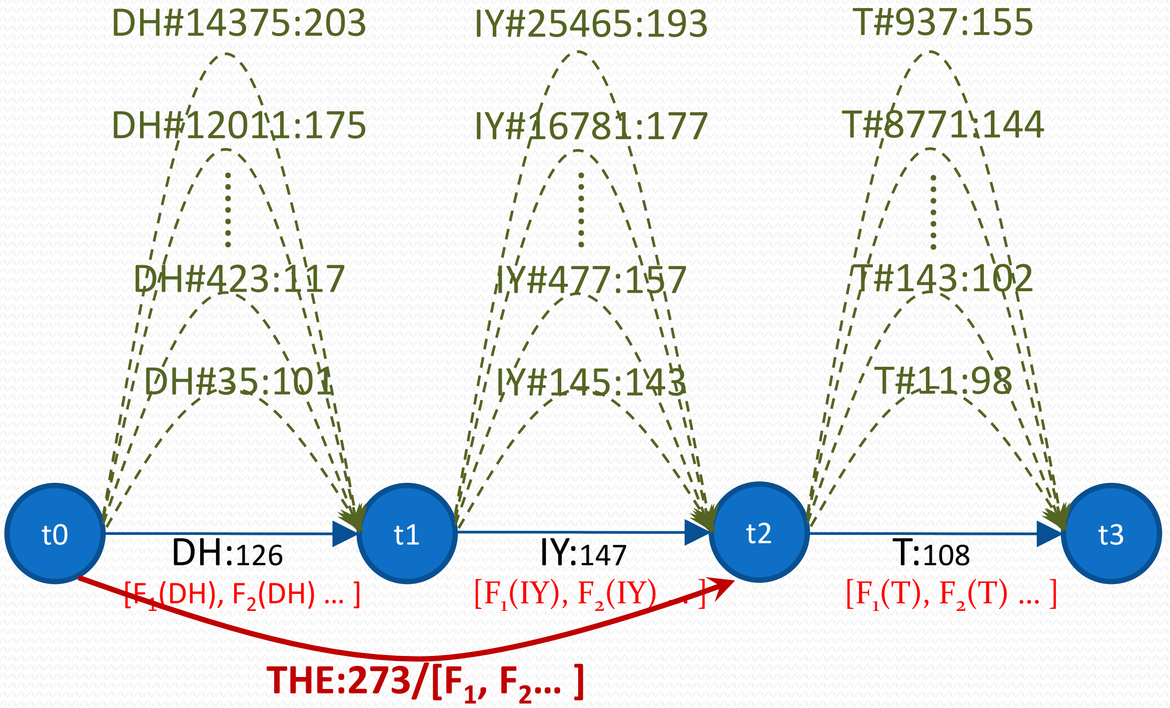
- words are the basic unit in SCARF

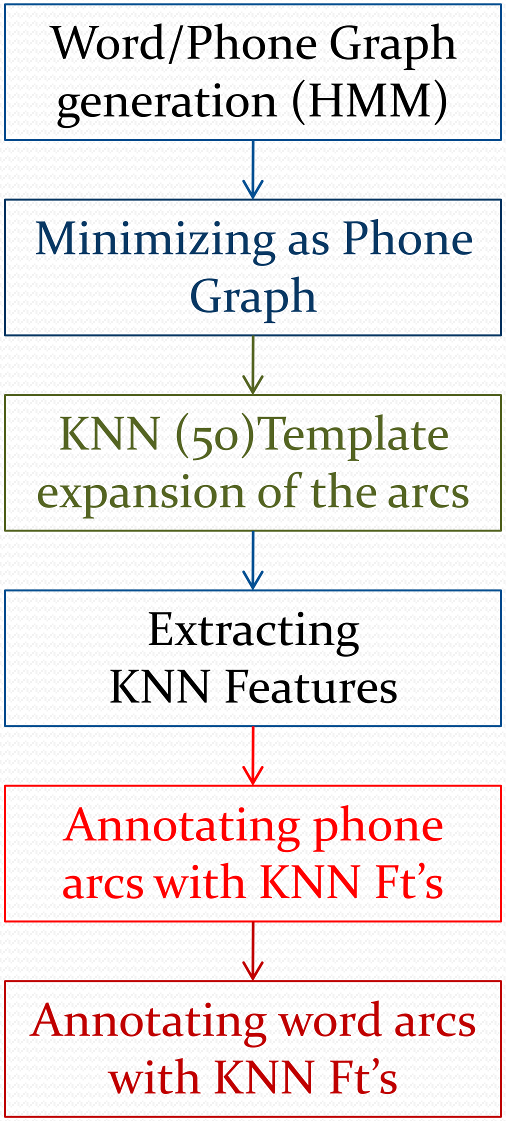
SUB structure: phone graph with score annotation

- phones are used as units in the template system for further processing

CONSTRAINTS: word arcs are unique taking cross-word context dependency into account

# Template Expansion and Feature Annotation





Word/Phone Graph generation (HMM)

Minimizing as Phone Graph

KNN (50)Template expansion of the arcs

Extracting KNN Features

Annotating phone arcs with KNN Ft's

Annotating word arcs with KNN Ft's

$$F_l(ph) = \sum_i P(T_i) F_l(T_i) \sim \sum_i \exp\left(-0.5 D_i\right) F_l(T_i)$$

$$F_l(wo) = \sum_{ph(wo)} F_l(ph(wo))$$

# Features Added at Workshop

- Word ID:
  - did the template originate from the same word ?
- Position Dependency (PD):
  - word initial, word final
  - having it as a feature favorably impacts granularity of the CD phone models vs. having CD and PD phones
- Averaged Score
  - Top-5 weighted average score
- Speaker ID entropy
  - it's taken as positive evidence that multiple speakers contribute to the KNN list
- Boundary Scores
  - How good is the match just beyond the boundaries of the current segment?
- Path constraints
  - fraction of non-diagonal moves in the DTW

# WSJ setup & HMM Baseline

- WSJ0+1 database:
  - 81hrs, 284 speakers
  - 644k words

- HMM Reference system:
  - feature extraction: mel spectra, VTLN, mean-norm
  - feature shaping: phone based MIDA (Mut. Info. DA)
  - shared pool of 32k gaussians components
  - 5875 cross-word CD triphones using on avg. 94 components
  - WER: 7.27 %
  - multiple variants in feature extraction and feature shaping (all in the range 7.27…7.58% WER)

# Template System - pre-workshop

- WSJ0+1 database:
  - 81hrs, 284 spkrs
  - 2.8 M phone templates

- Implementation choices:
  - ~ 5k CD phone classes
  - feature extraction: cfr. HMM
  - single best decoding
  - WER: 9.8 %

# Template System (New Results)

- Pre-Workshop WER: 9.80%
- Improved implementation: ~ 10% relative better
- Contributions in the SCARF framework: ~ 10% relative
  - Word ID:
  - Position Dependency:
  - Improved KNN List Generation:
  - Speaker ID entropy:
  - Averaged Score:
  - Path constraints:
  - Signal Continuity Score:
- Combined System: 8.1%

# System Combination Results

| **Wall Street Journal** | **WER** |
|---|---|
| Template System pre-workshop | 9.8 % |
| Template System  DTW score only | 9.1 % |
| + SCARF, multiple features | 8.1 % |
| Baseline HMM | 7.3 % |
| + SCARF, phone detectors | 6.8 % |
| + SCARF, template features and phone detectors | **6.7 %** |
| (Lattice Oracle – best achievable) | 2.9 % |

# Broadcast News Fundamentals

Damianos Karakos

# The BN Corpus

- Training Data
  - 430 hours of audio (HUB4)
  - ~5 million words
- Development Data (Dev04f)
  - 2 hours (Dev04f)
  - ~22K words
- Test Data
  - 4 hours (RT04f)
  - ~50K words

# Attila Baseline

- Attila: state-of-the-art speech recognizer by IBM
- Based on Hidden Markov Models with Gaussian mixtures
- Consists of a series of steps:
  - Maximum Likelihood + Linear Discriminant Analysis
  - Vocal Tract Length Normalization
  - Speaker-adapted training (MLLR and fMLLR)
  - Discriminative training (Boosted MMI)

# Attila Baseline Error Rates

All the standard methods are in it

| | Dev04f WER | RT04f WER |
|---|---|---|
| ML + LDA | 30.6% | 28.4% |
| + VTLN | 23.3 | 21.9 |
| + fMLLR | 21.2 | 20.3 |
| + MLLR | 20.5 | 19.8 |
| + fMMI | 17.0 | 16.3 |
| + mMMI | 16.5 | 15.9 |
| + open beams | 16.3 | 15.7 |

Gains from some standard methods ~1%

# SCARF Baseline Error Rates

- Attila (IBM recognizer) output was used as the "baseline feature" of SCARF.
    - Time-annotated word string.
    - Essentially a discretized AM score
- Provides a "safety net" for SCARF

| | Dev04f WER |
|---|---|
| Attila Baseline | 16.3% |
| SCARF with baseline | 16.0 |

# SCARF Baseline Error Rates

- Attila (IBM recognizer) output was used as the "baseline feature" of SCARF.
  - Time-annotated word string.
  - Essentially a discretized AM score
- Provides a "safety net" for SCARF

| | Dev04f WER |
|---|---|
| Attila Baseline | 16.3% |
| **SCARF1** | 16.0 |

# Adding MSR Word Detectors

| | Dev04f WER |
|---|---|
| Attila Baseline | 16.3% |
| SCARF1 | 16.0 |
| + **MSR Word Detectors** | 15.3 |

This system often referred to in later talks.

# Language Modeling and Word Detection Experiments

Damianos Karakos

# Experiments with SCARF

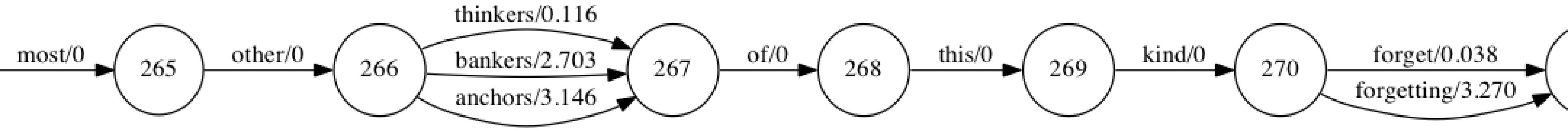
- **Key Research**
  - Cohort set based detections
- **Comparison with ROVER**
  - Contrastive Attila systems for ROVER: (i) with triphone decision tree, (ii) with reduced question set.
  - ROVER did not exploit the information sources
- **Comparison with LM Rescoring**
  - SCARF exploited multiple LMs effectively

# Experiments with SCARF

- **Key Research**
  - Cohort set based detections
- **Comparison with ROVER**
  - Contrastive Attila systems for ROVER: (i) with triphone decision tree, (ii) with reduced question set.
  - ROVER did not exploit the information sources
- **Comparison with LM Rescoring**
  - SCARF exploited multiple LMs effectively

# Cohort-set based detectors

- Cohort set of a word *w*: the set of words which are found frequently confused with *w* in the training data (or some other untranscribed corpus).
- *Confusion networks* can be used to compute cohort sets.



# Examples of cohort sets

- **accept** except (152) accepted (22) accepts (18) accepting (5) exit (4) expect (3) set (2) exception (2) …
- **party's** parties (139) party (31) parties' (30) part (4) authorities (4) partisan (2) …
- **tails** tales (22) details (6) talese (6) tells (5) entails (3) sales (2) tail (2) hills (2) tailed (2) tale (2) motels …
- **yield** field (9) deal (6) feel (4) yields (3) heeled (3) sealed (3) deals (3) healed (3) appealed (3) know (2) yielded (2) guild (2) heal (2) reveal (2) …
- …

# Using cohorts to build word detectors

- For each word *w* we built a binary classifier (detector) using n-gram features.
- The classifier of *w* gives the probability that the word following a n-gram history is *w*.
- Training data: all occurrences of *w* in the language modeling text (BN corpus) and *all occurrences of its cohort words.*

# Example

... THE TWO THOUSAND **ELECTION** CYCLE ...
... GOING TO BE AN **ELECTION** IN ...
... HERE TO AN **ELECTION** IS ...
... MEMBERS SHOW UP ON **ELECTION** DAY ...

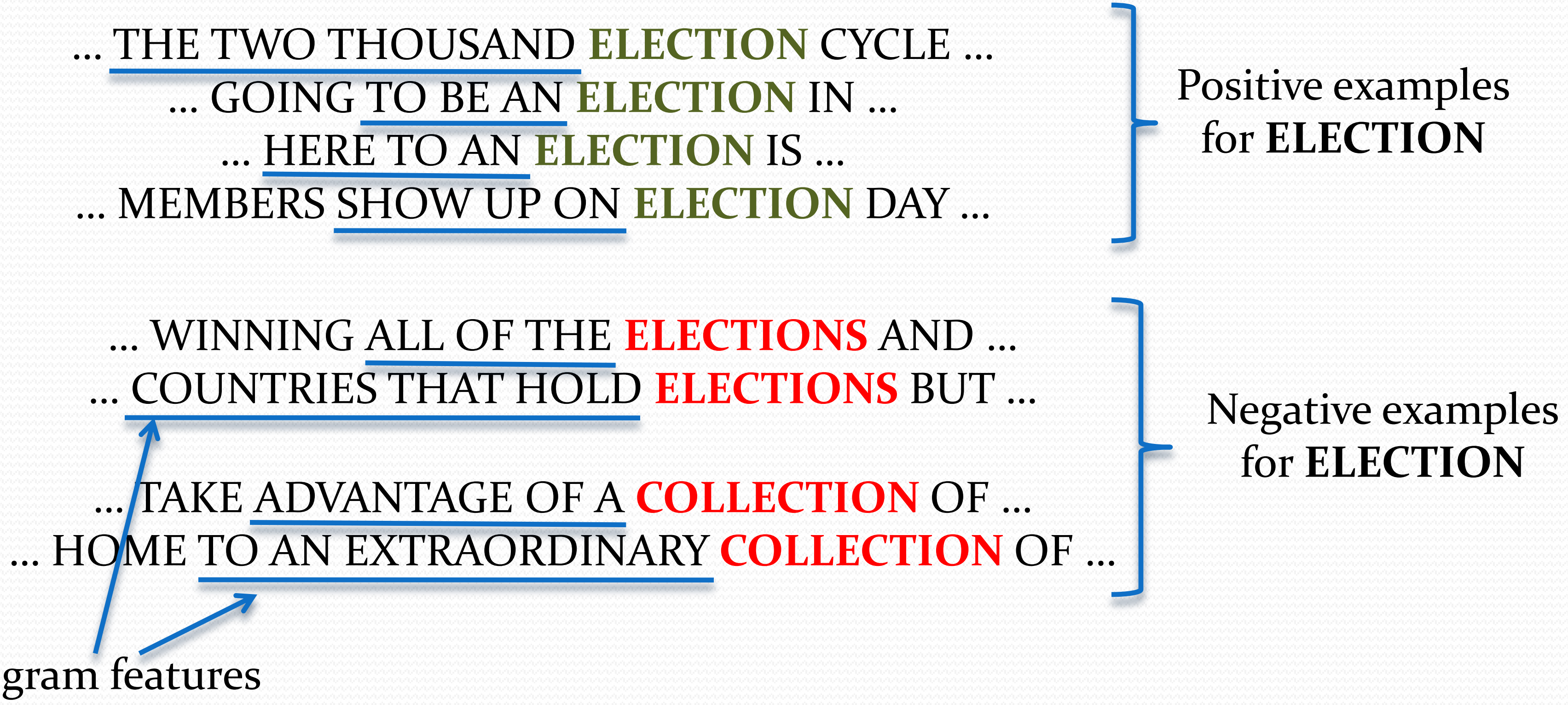
Positive examples for **ELECTION**

... WINNING ALL OF THE **ELECTIONS** AND ...
... COUNTRIES THAT HOLD **ELECTIONS** BUT ...

... TAKE ADVANTAGE OF A **COLLECTION** OF ...
... HOME TO AN EXTRAORDINARY **COLLECTION** OF ...

Negative examples for **ELECTION**

# Example



... THE TWO THOUSAND **ELECTION** CYCLE ...
... GOING TO BE AN **ELECTION** IN ...
... HERE TO AN **ELECTION** IS ...
... MEMBERS SHOW UP ON **ELECTION** DAY ...

Positive examples for **ELECTION**

... WINNING ALL OF THE **ELECTIONS** AND ...
... COUNTRIES THAT HOLD **ELECTIONS** BUT ...

... TAKE ADVANTAGE OF A **COLLECTION** OF ...
... HOME TO AN EXTRAORDINARY **COLLECTION** OF ...

Negative examples for **ELECTION**

3-gram features

Used a max-ent classifier (developed by P. Nguyen)

# Using cohorts to build word detectors

- At any particular position in the lattice (confusion network), apply the detectors for all words in competition → binary features for SCARF.
- Note: we only focus on non-function word confusions.

1185 1227 MAYORS f1=1,f2=0
1185 1228 MAYORS f1=1,f2=0
1228 1247 AND f1=1,f2=0
1229 1246 AND f1=1,f2=0
1247 1275 TOWN f1=1,f2=0
1248 1276 TOWN f1=1,f2=0
1276 1323 COUNCIL f1=1,f2=0
1277 1322 COUNCIL f1=1,f2=0
1323 1373 MEMBERS **f1=1**,f2=0
1323 1376 MEMBERS **f1=1**,f2=0
1323 1376 MEMBERS' f1=0,**f2=-1**
1324 1376 MEMBERS **f1=1**,f2=0

240 261 HELD f1=1,f2=0
262 289 KEY f1=1,f2=0
263 290 KEY f1=1,f2=0
290 327 LOCAL f1=1,f2=0
291 327 LOCAL f1=1,f2=0
328 340 AND f1=1,f2=0
341 388 PROVINCIAL f1=1,f2=0
341 389 PROVINCIAL f1=1,f2=0
389 439 ELECTION f1=0,**f2=-1**
389 443 ELECTIONS **f1=1**,f2=0
390 443 ELECTIONS **f1=1**,f2=0
440 491 SUNDAY f1=1,f2=0

# Results

Discard baseline feature to emphasize language model

| | Without word-det | With word-det |
|---|---|---|
| SCARF with 1-gram | 21.3 | 19.0 |
| SCARF with 2-gram | 19.2 | 18.4 |
| SCARF with 3-gram | 17.8 | 17.7 |

- Consistent gain from using cohort based detectors
- Good results from training with lattice confusions also observed in later talk by Aren

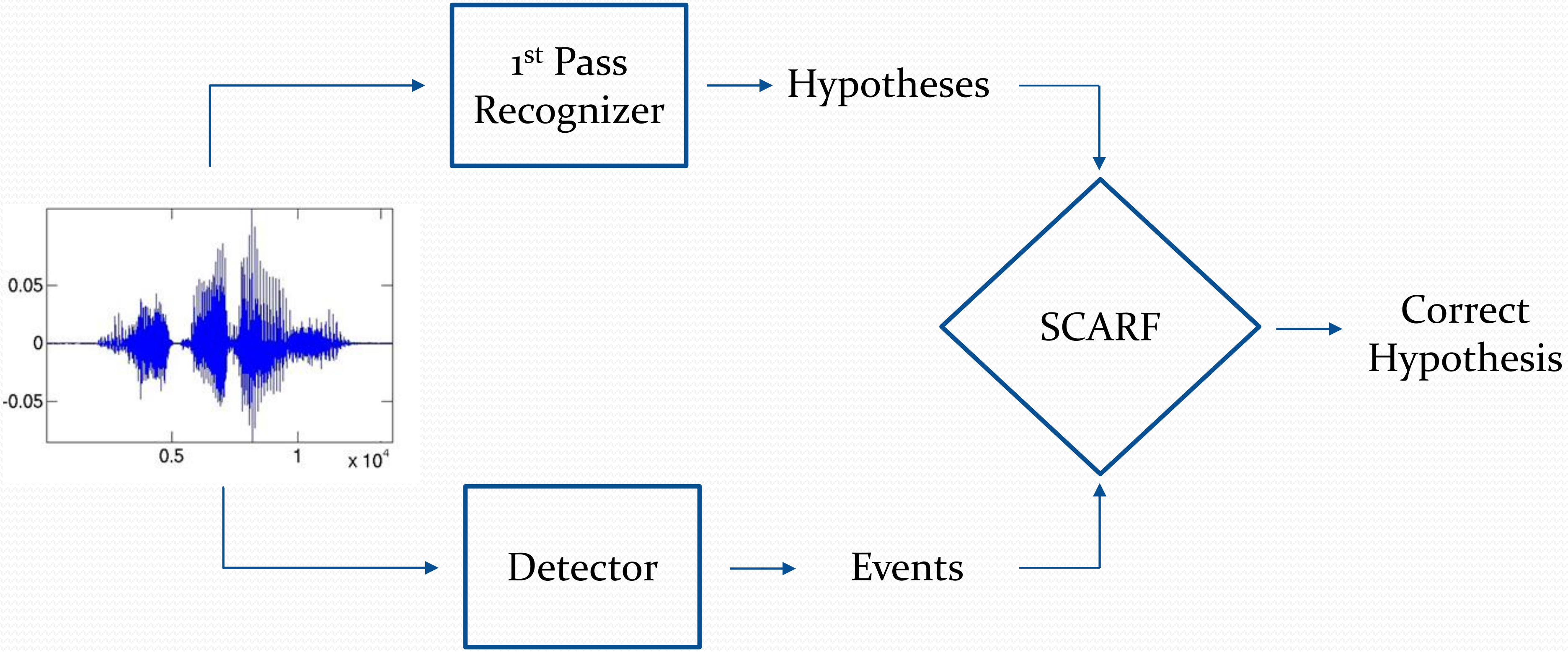
# Detecting Phonetic Events in Speech & MLP based Posteriors

Samuel Thomas

Sivaram GSVS

Hynek Hermansky

# Detecting Phonetic Events



1st Pass Recognizer

Hypotheses

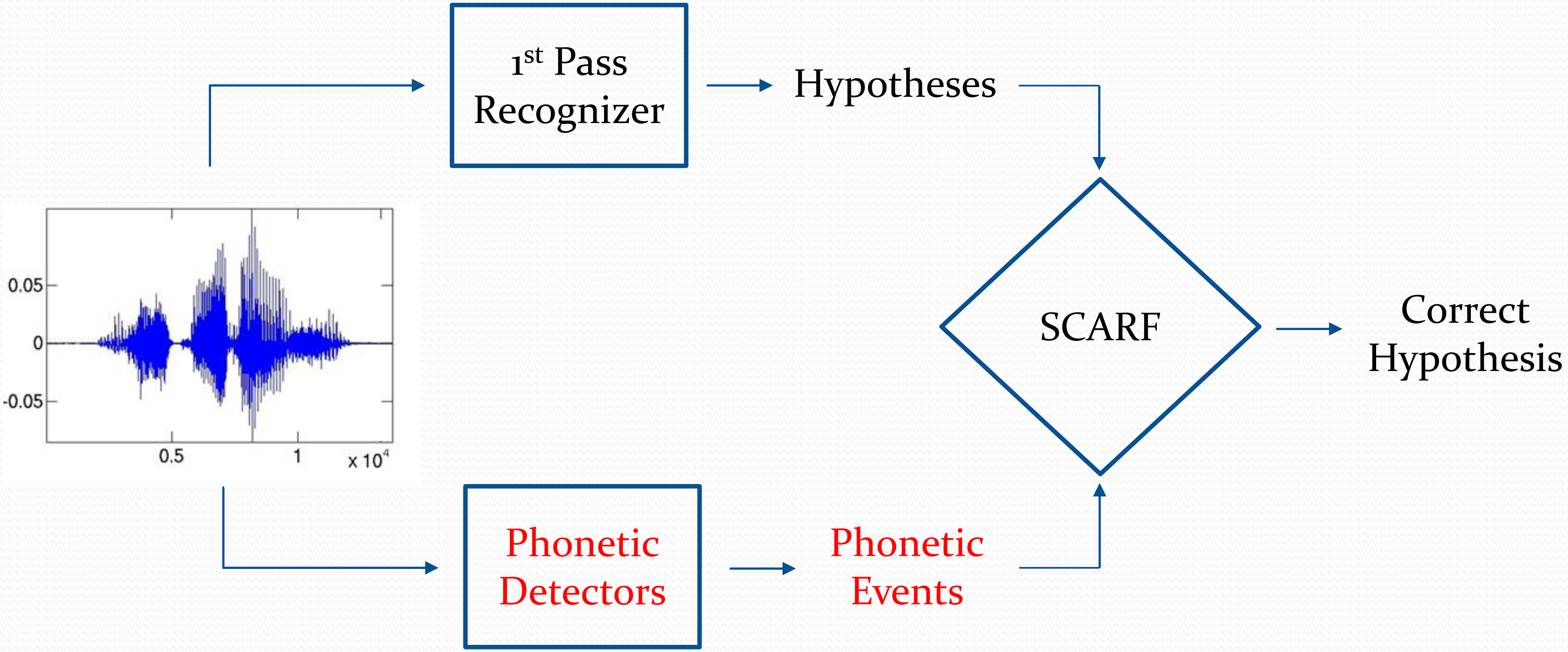
SCARF

Correct Hypothesis

Detector

Events

# Detecting Phonetic Events



1st Pass Recognizer

Hypotheses

SCARF

Correct Hypothesis

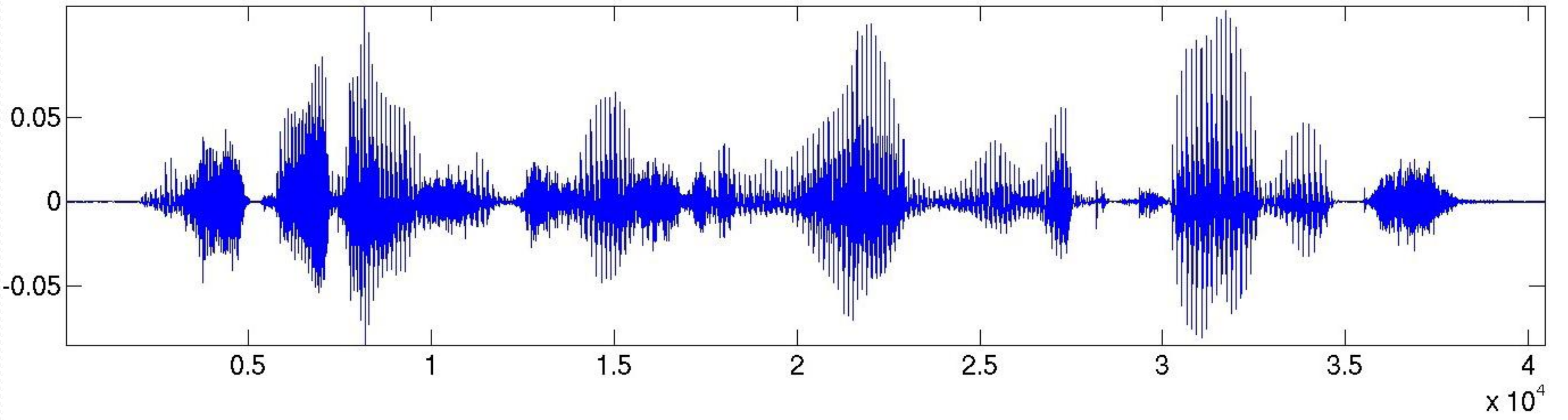
Phonetic Detectors

Phonetic Events

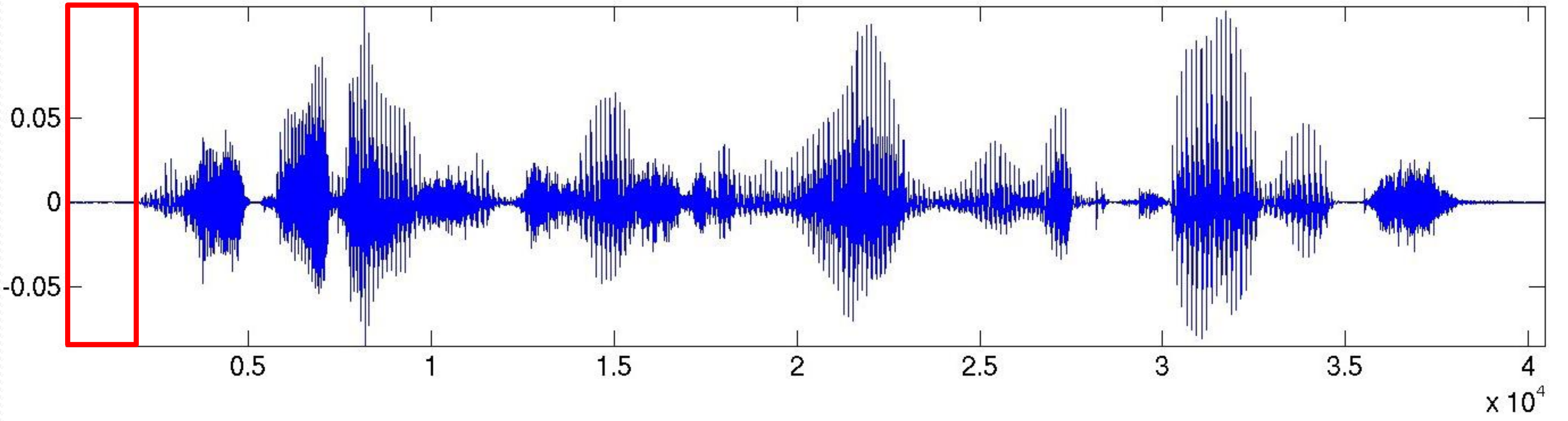
# How do we build phonetic detectors?

- Phoneme recognizers using posteriors from MLPs
- Phoneme recognizers from Deep NNs (next talk)

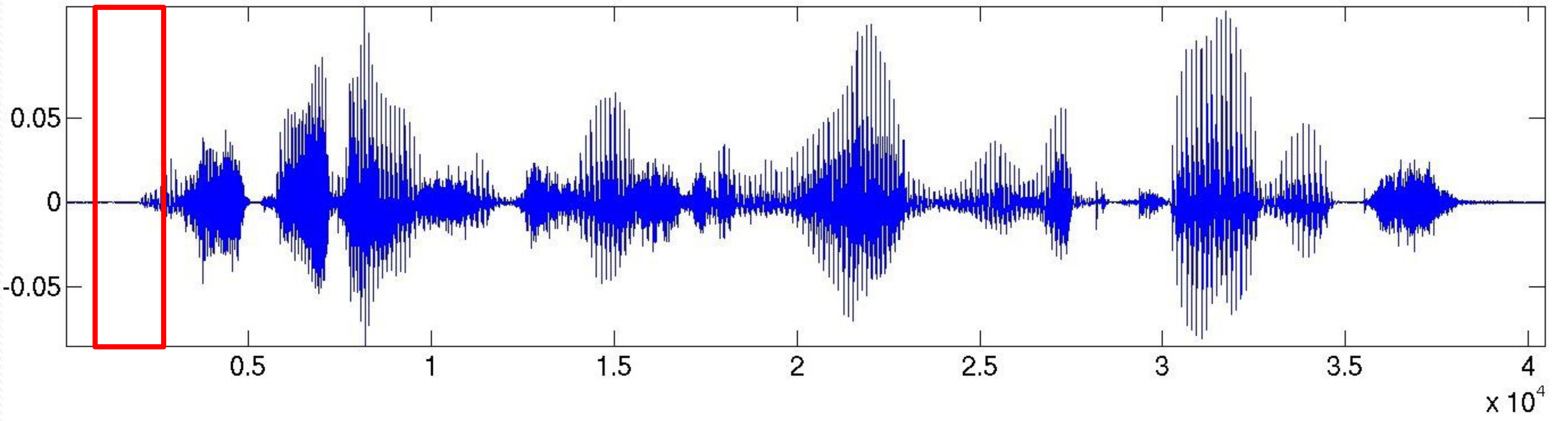
# Detecting Phonetic Events



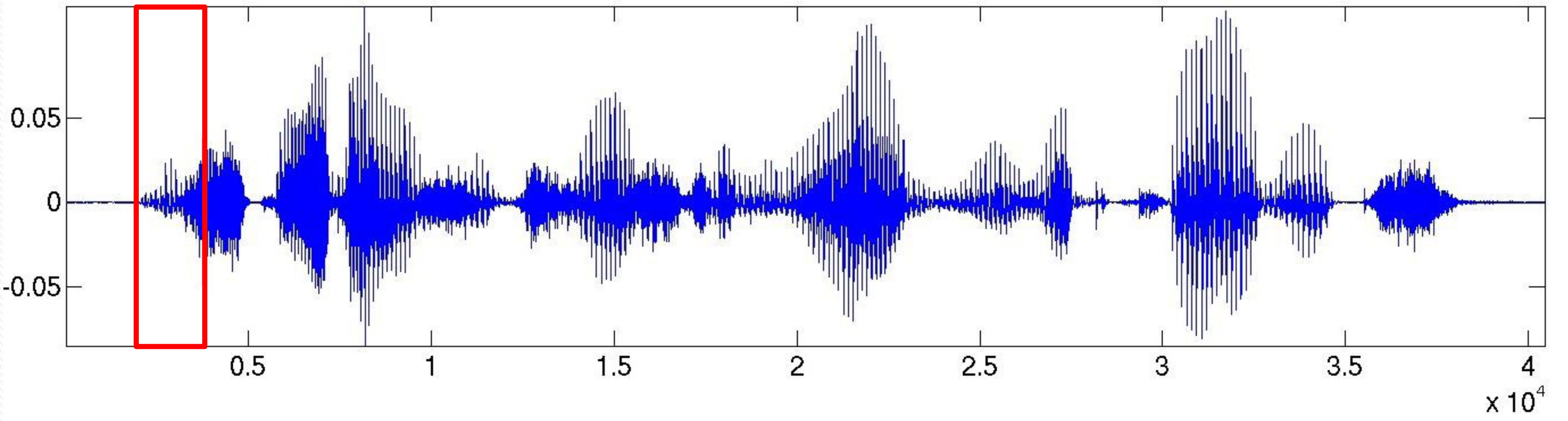
# Detecting Phonetic Events



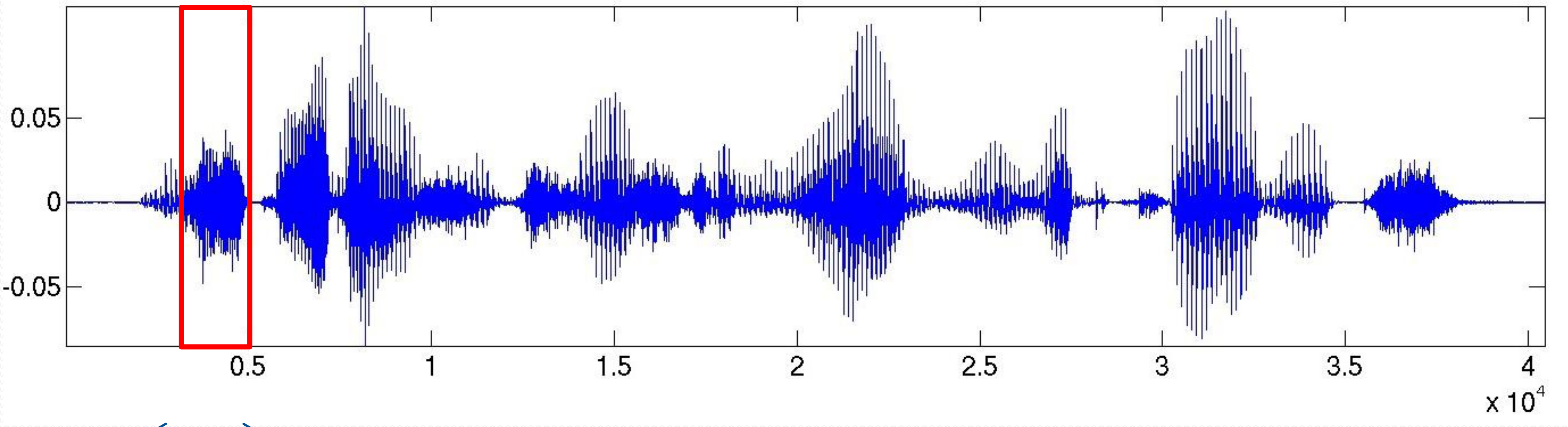
# Detecting Phonetic Events



# Detecting Phonetic Events

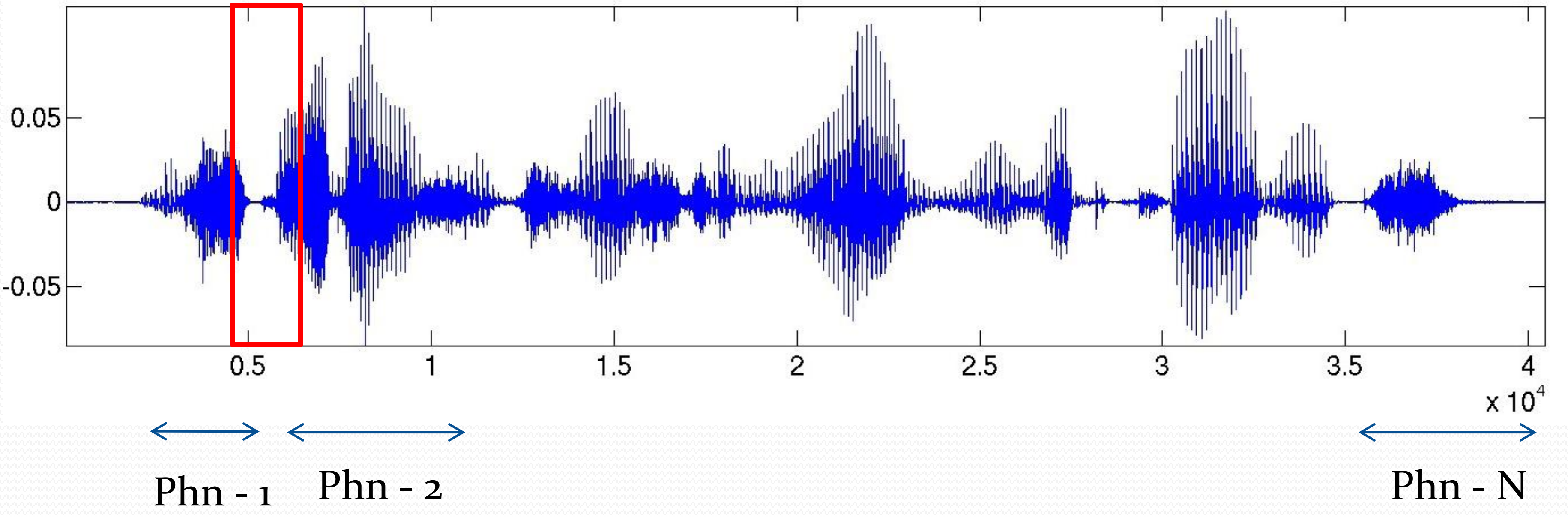


# Detecting Phonetic Events

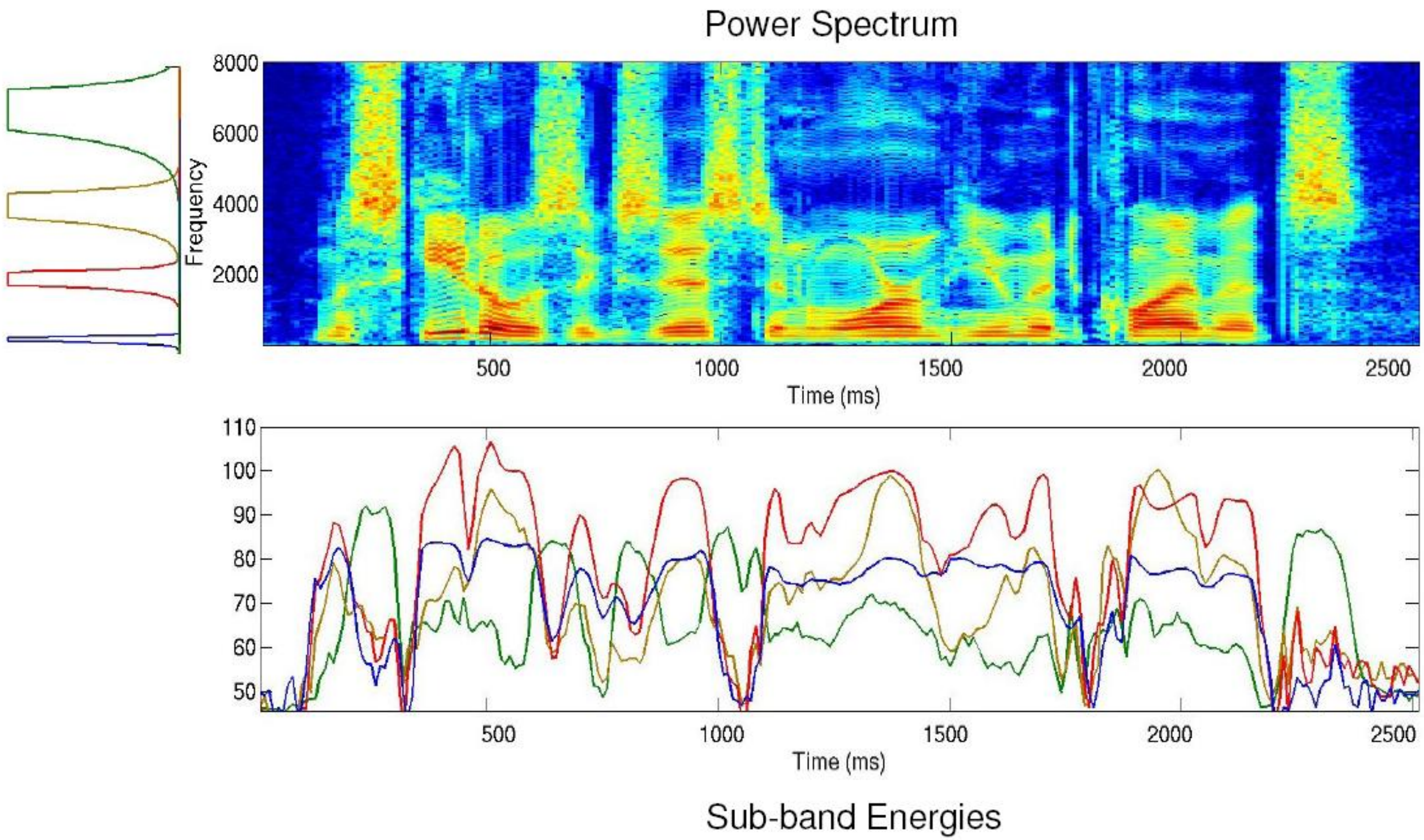


Phn - 1

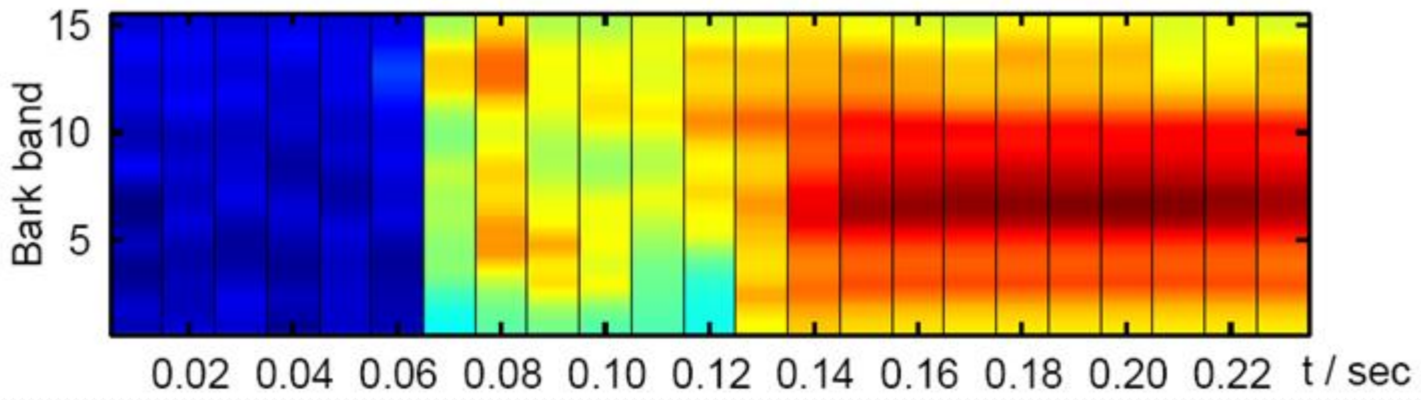
# Detecting Phonetic Events



# Detecting Phonetic Events

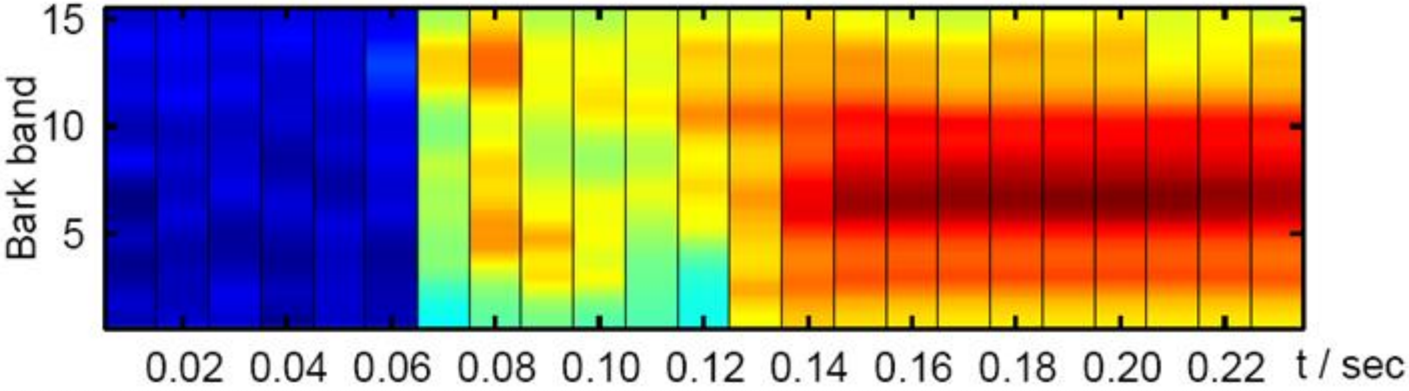


# Detecting Phonetic Events

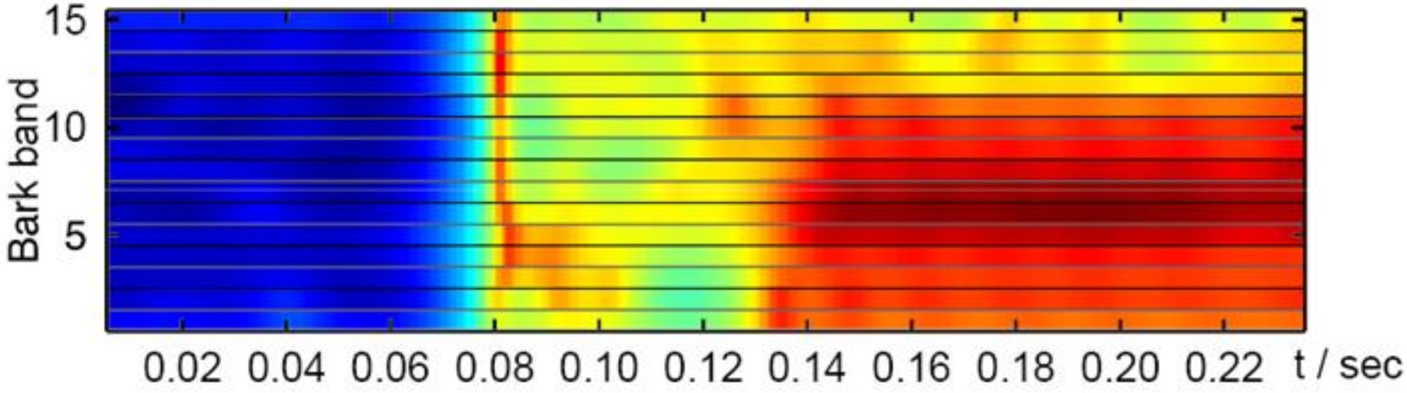


PLP (Perceptual Linear Prediction)
- Conventional Feature Extraction Techniques

# Detecting Phonetic Events

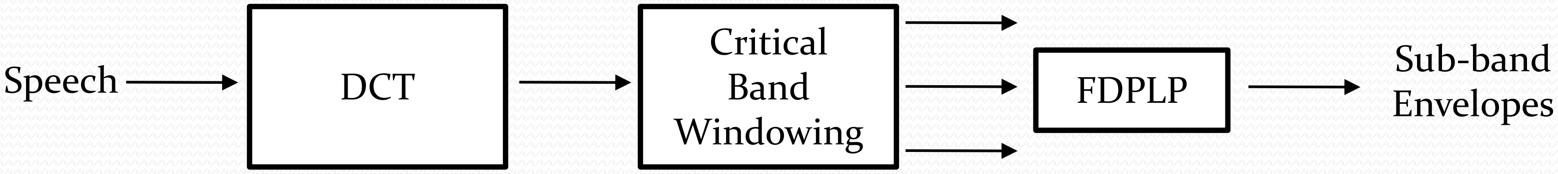


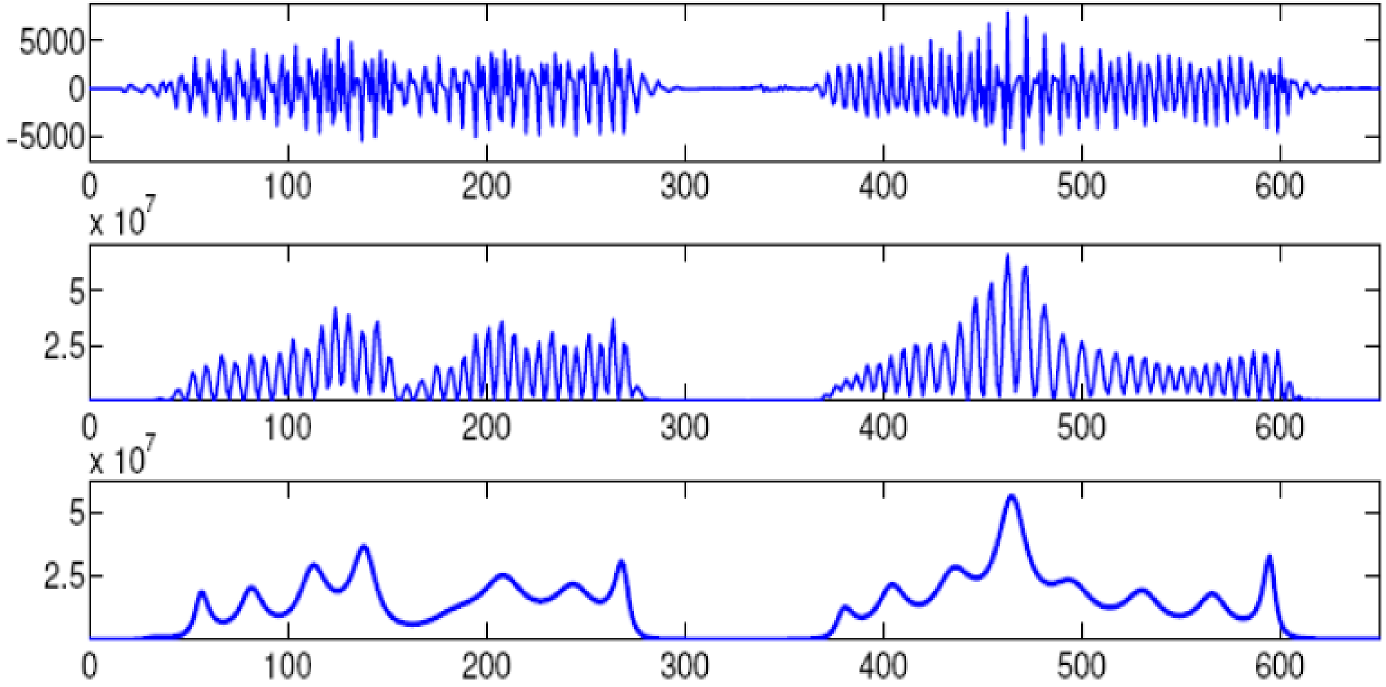
PLP (Perceptual Linear Prediction)
- Conventional Feature Extraction Techniques



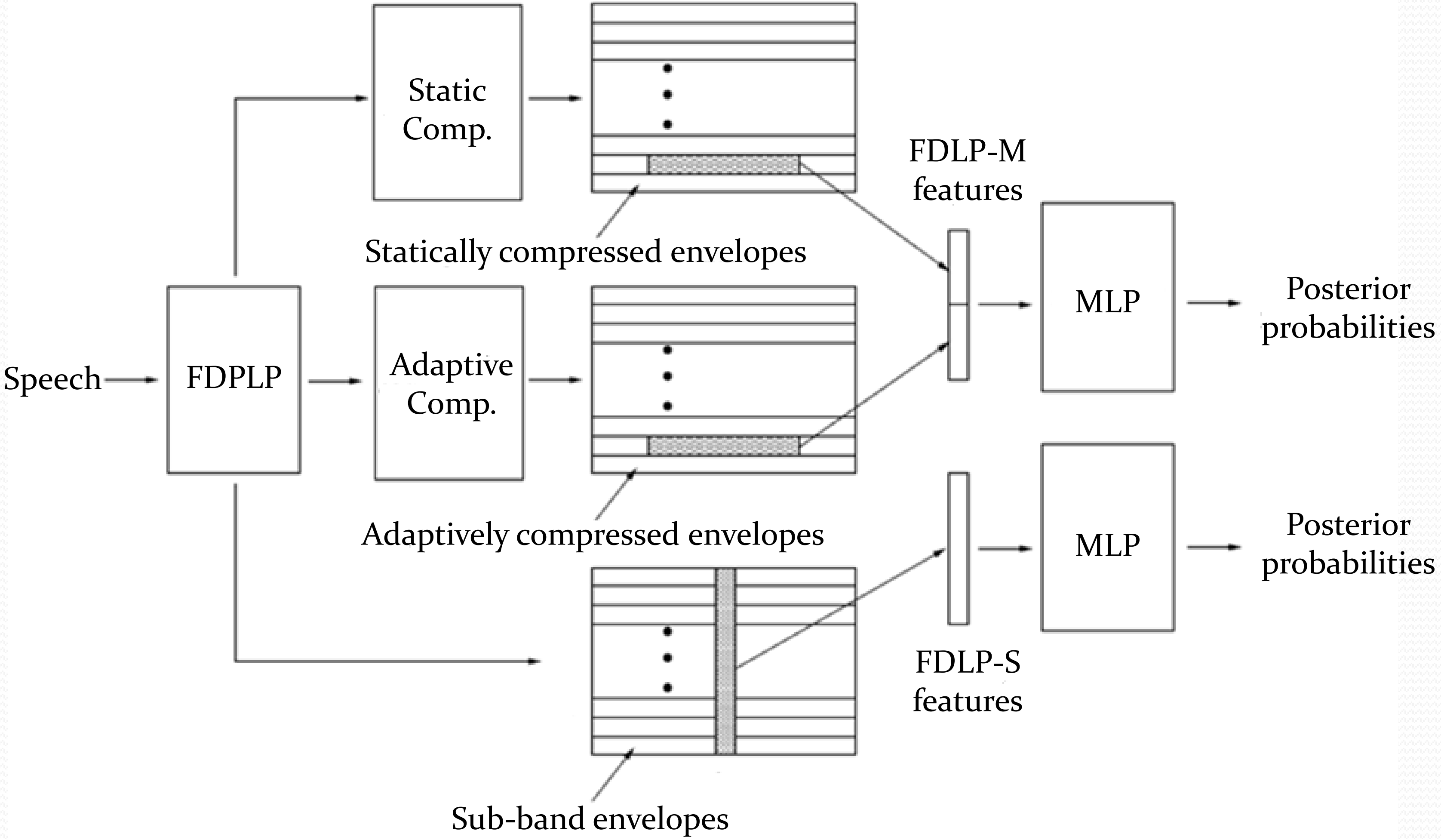
FDPLP (Frequency Domain Perceptual Linear Prediction)

# Detecting Phonetic Events

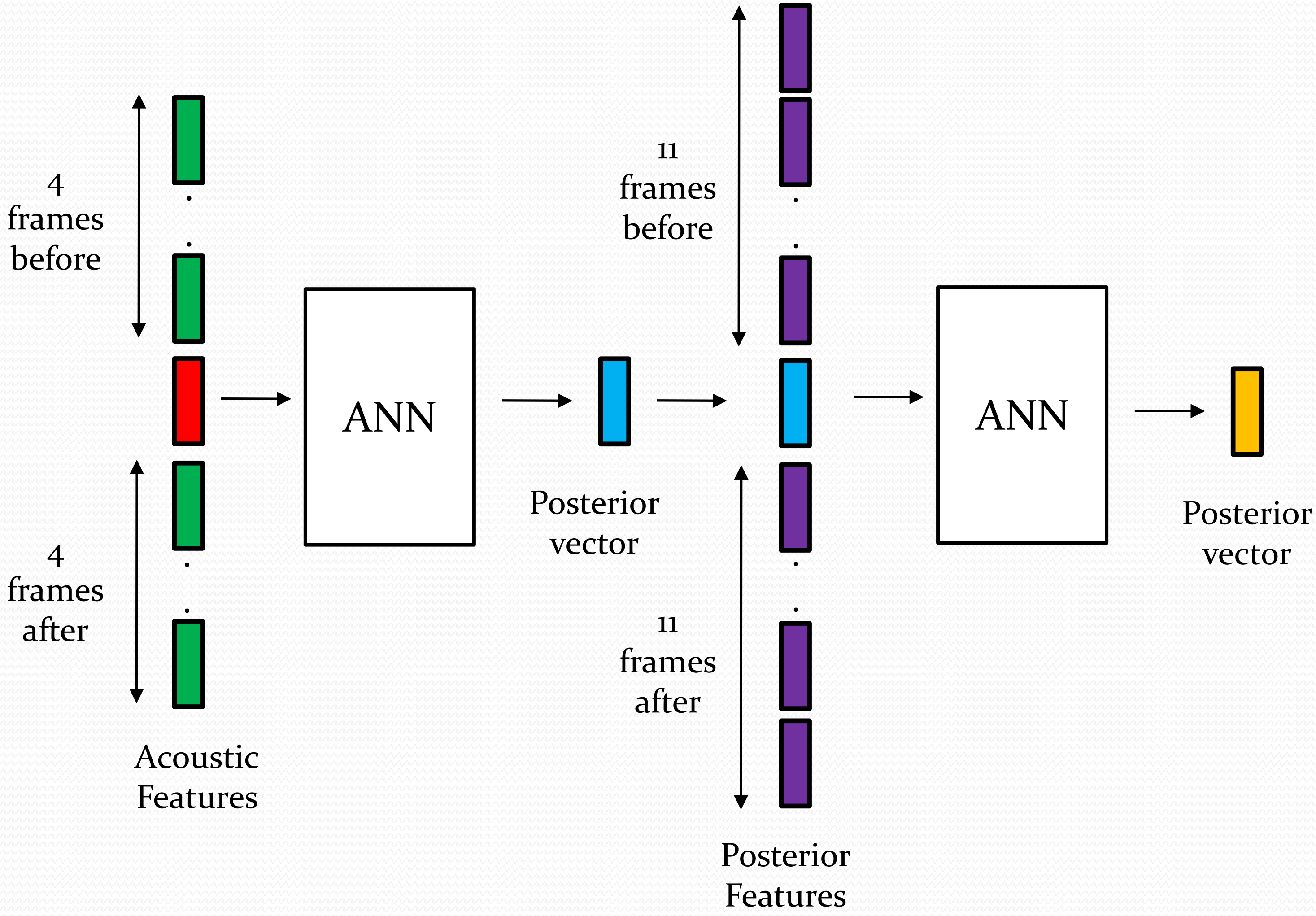




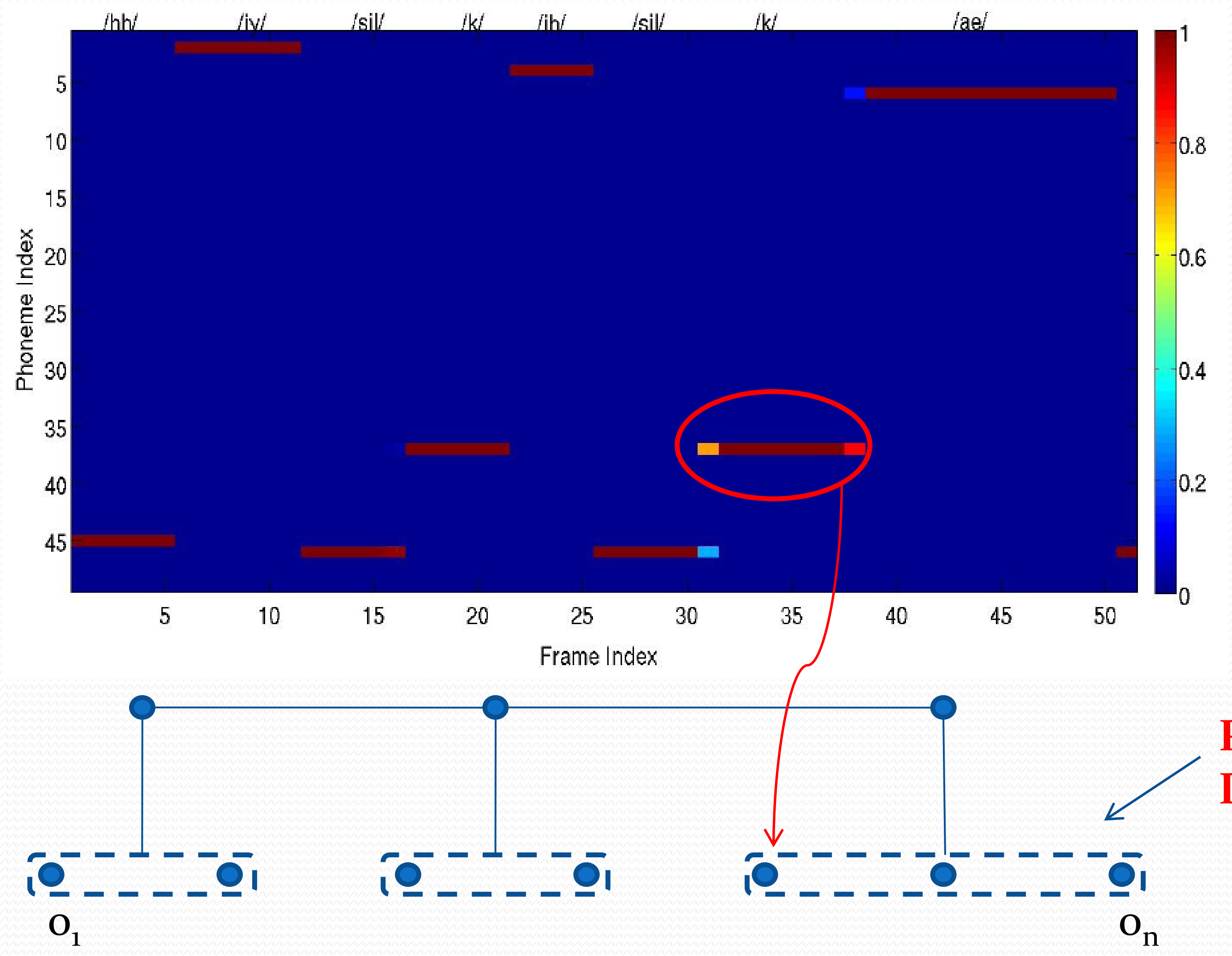
# Detecting Phonetic Events



# Detecting Phonetic Events

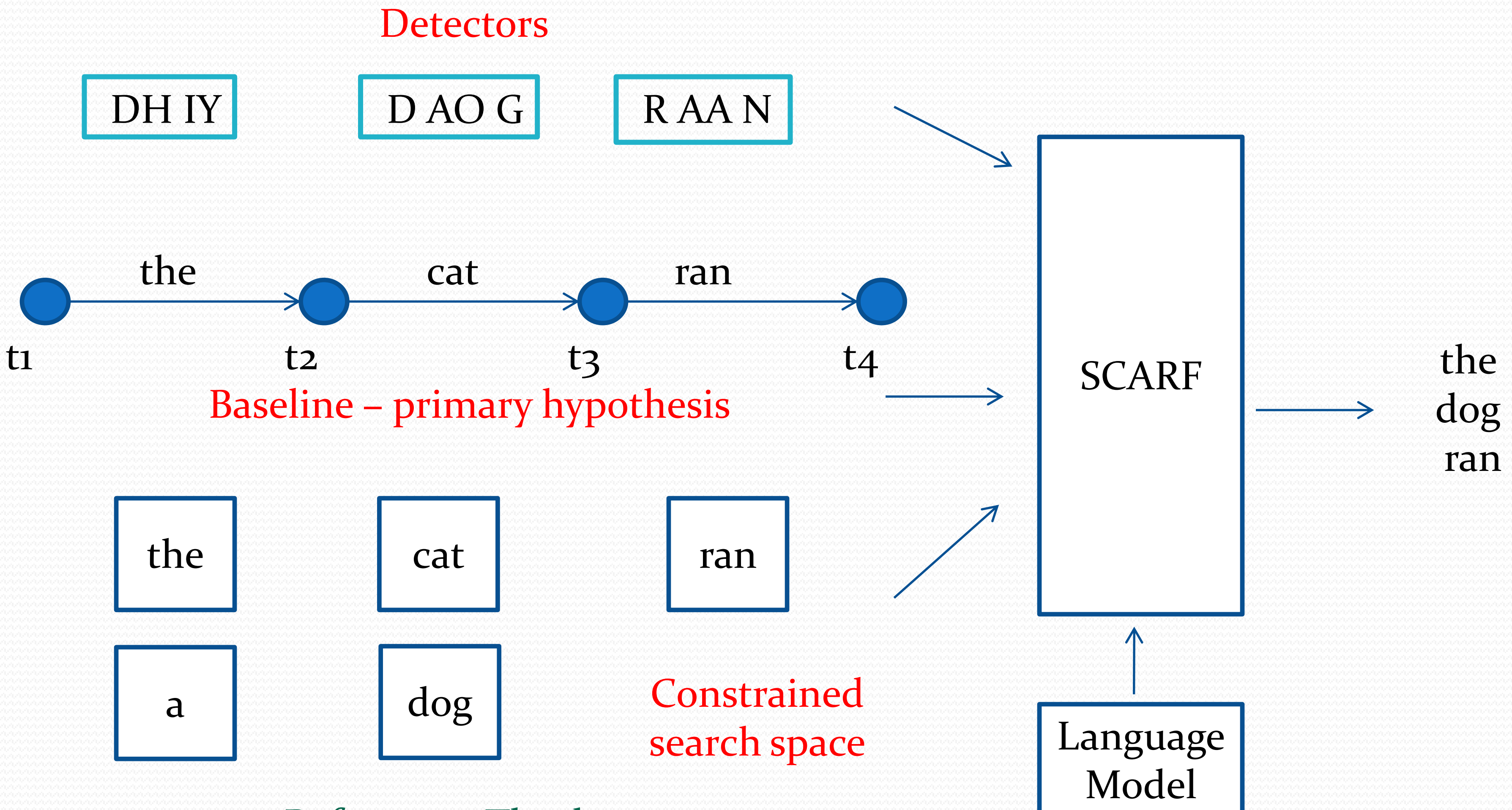


# Detecting Phonetic Events



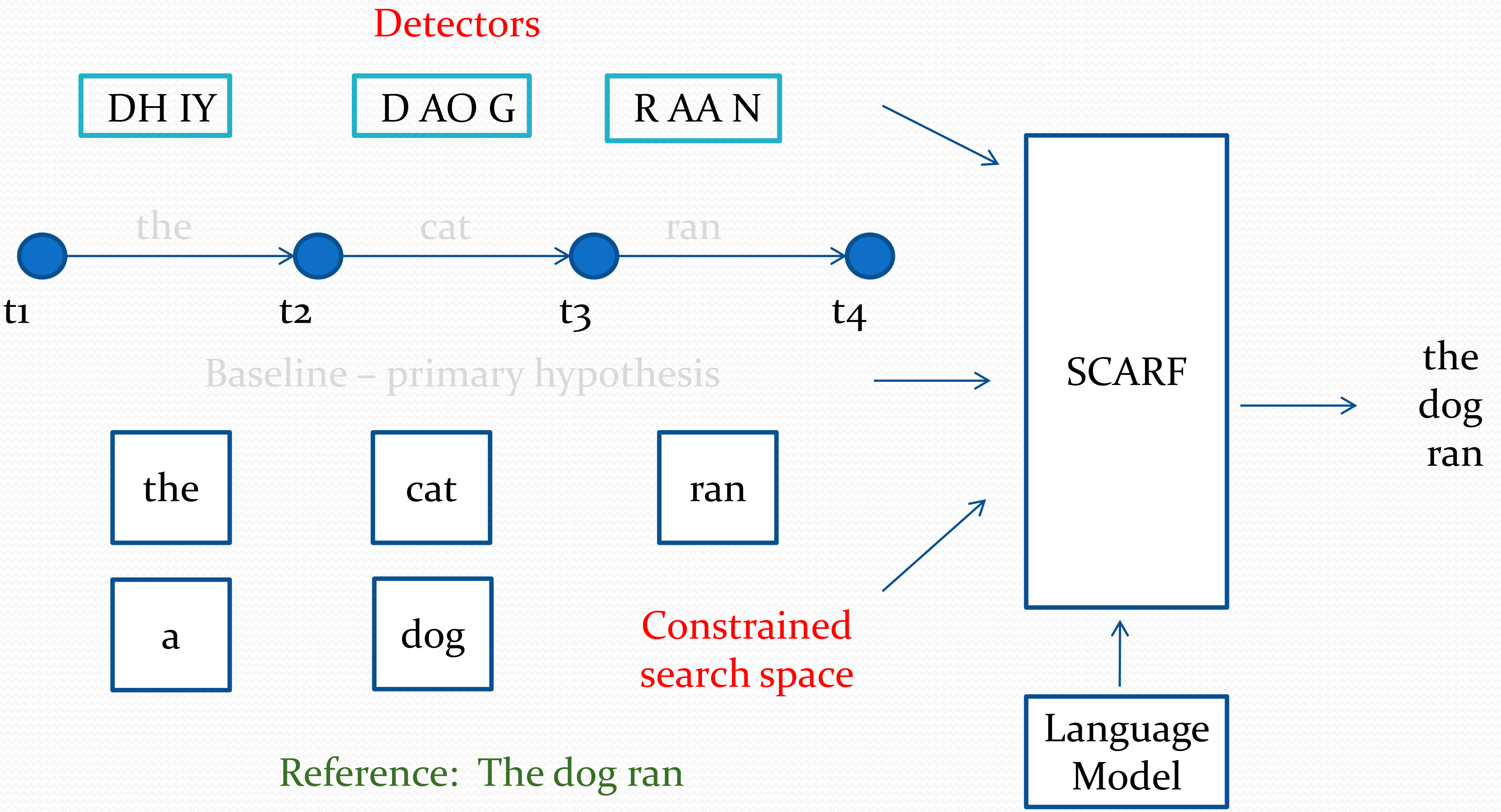
**Phoneme Detections**

# Putting everything into SCARF



Reference: The dog ran

# Putting everything into SCARF



Detectors

DH IY

D AO G

R AA N

the

cat

ran

t1

t2

t3

t4

Baseline – primary hypothesis

the

cat

ran

a

dog

Constrained search space

SCARF

the dog ran

Language Model

Reference: The dog ran

# Phoneme Detectors as Acoustic models

| Acoustic Information | PER | WER |
|---|---|---|
| None | -- | 17.9% |
| Perceptual Linear Prediction (PLP) | 32.5% | 17.2% |
| PLP-Sparse | 31.0% | 17.3% |
| **FDLP-S** | **31.1%** | **17.0%** |
| **FDLP-M** | **28.9%** | **16.9%** |

Phoneme detectors **capture information in the acoustic signal** – new feature extraction techniques **improve over conventional feature extraction techniques**

# Phoneme Detectors in Full System

| Acoustic Information | WER |
|---|---|
| SCARF1 + MSR | 15.3% |
| **+ MLP based Phoneme Detectors** | **15.1 %** |

MLP based phoneme detectors are able to **correct errors in the baseline hypothesis** and hence **decrease WERs**

# Summary

- We have investigated a new technique – **Frequency Domain Perceptual Linear Prediction (FDPLP)** to derive features for speech recognition
- Posteriors from MLPs have been traditionally integrated into LVCSR system using the TANDEM approach – We have now successfully integrated **posterior information as phoneme detectors** using SCARF
- Sharper posteriors derived using novel features have been used as **input to other acoustic modeling techniques - Point process models**

# Deep Neural Net Phoneme Detectors

Fei Sha

Meihong Wang

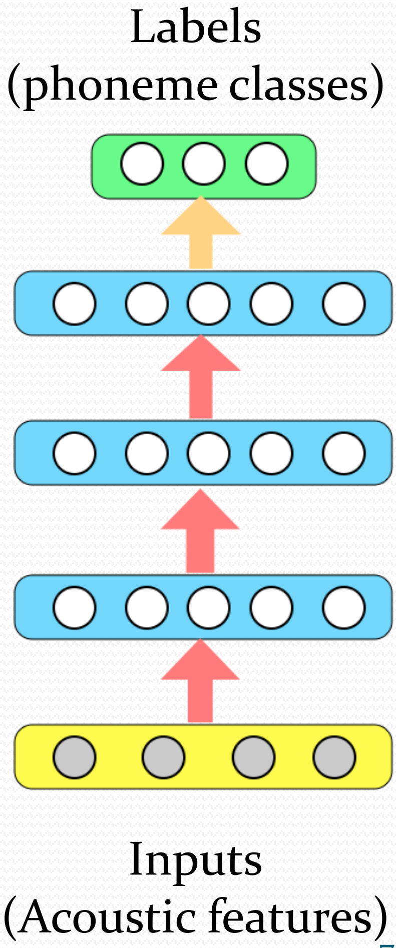
# Motivation

- Scientifically novel
  - Combining several contemporary ideas in machine learning: semi-supervised learning, regularization, stochastic optimization
- Empirically successful
  - Achieving state-of-the-art results: computer vision, natural language processing, phoneme recognition

**Goal: examine the utility of deep nets in standard large-vocabulary speech recognition**

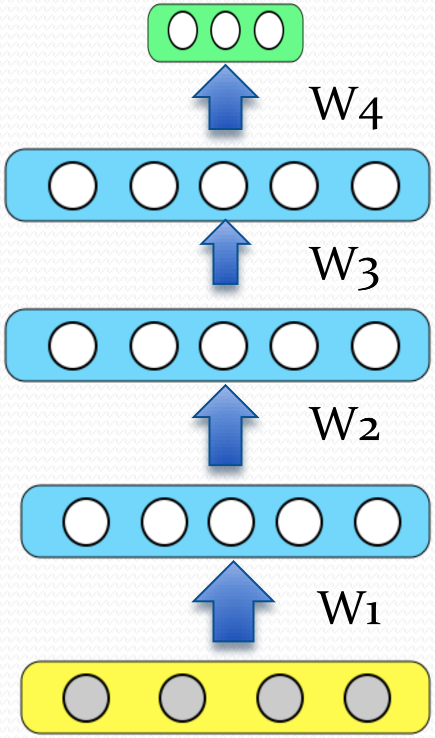
# Deep neural nets are

- Similar to multilayer perceptron
  - Propagate inputs through feed-forward layers
  - Compute posterior probabilities of categorical output variables



# Deep neural nets are

- Very different from multi-layer perceptron
  - Supervised globally, unsupervised locally



Supervised learning
(all weights adjusted)

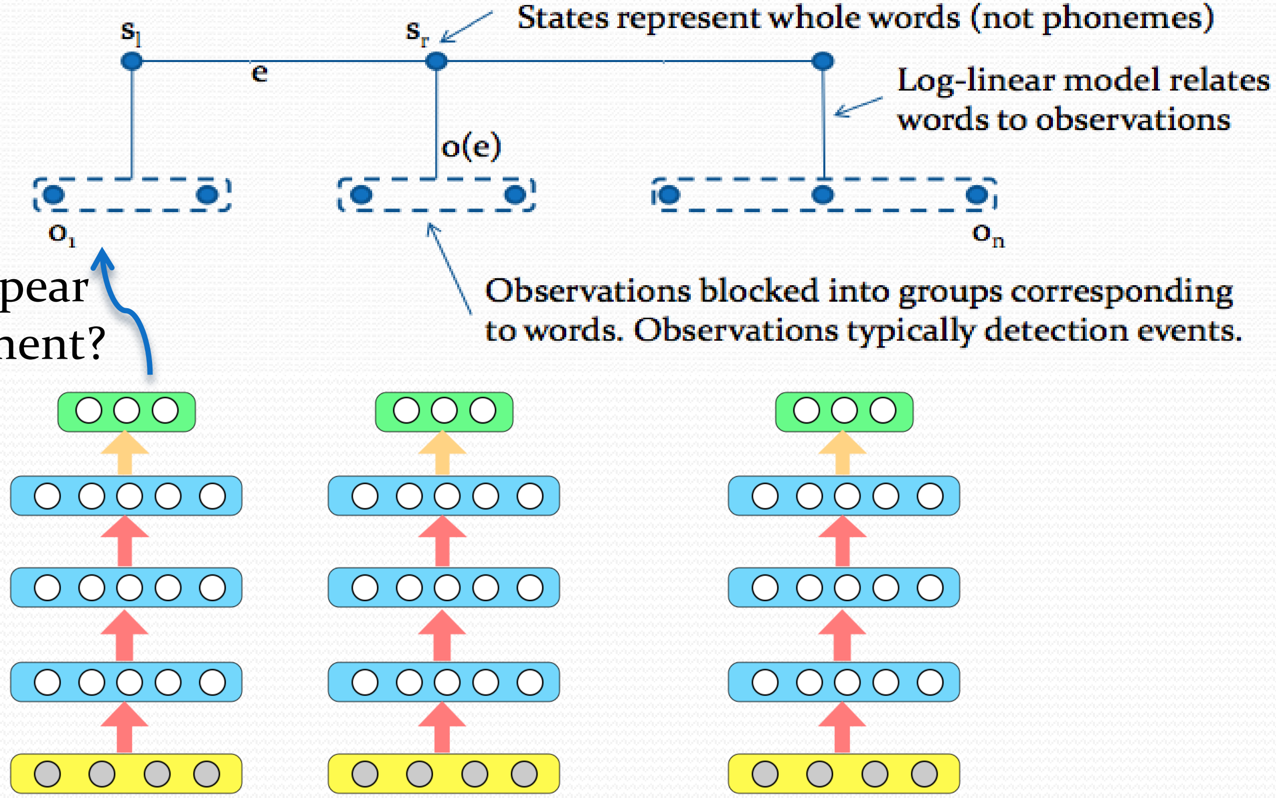
Unsupervised learning
(while fixing W1 and W2)

Unsupervised learning
(while fixing W1)

Unsupervised learning

# Apply deep nets to LVCSR, how?

- Build deep nets based phoneme detectors
- Leverage on SCARF to integrate detection results



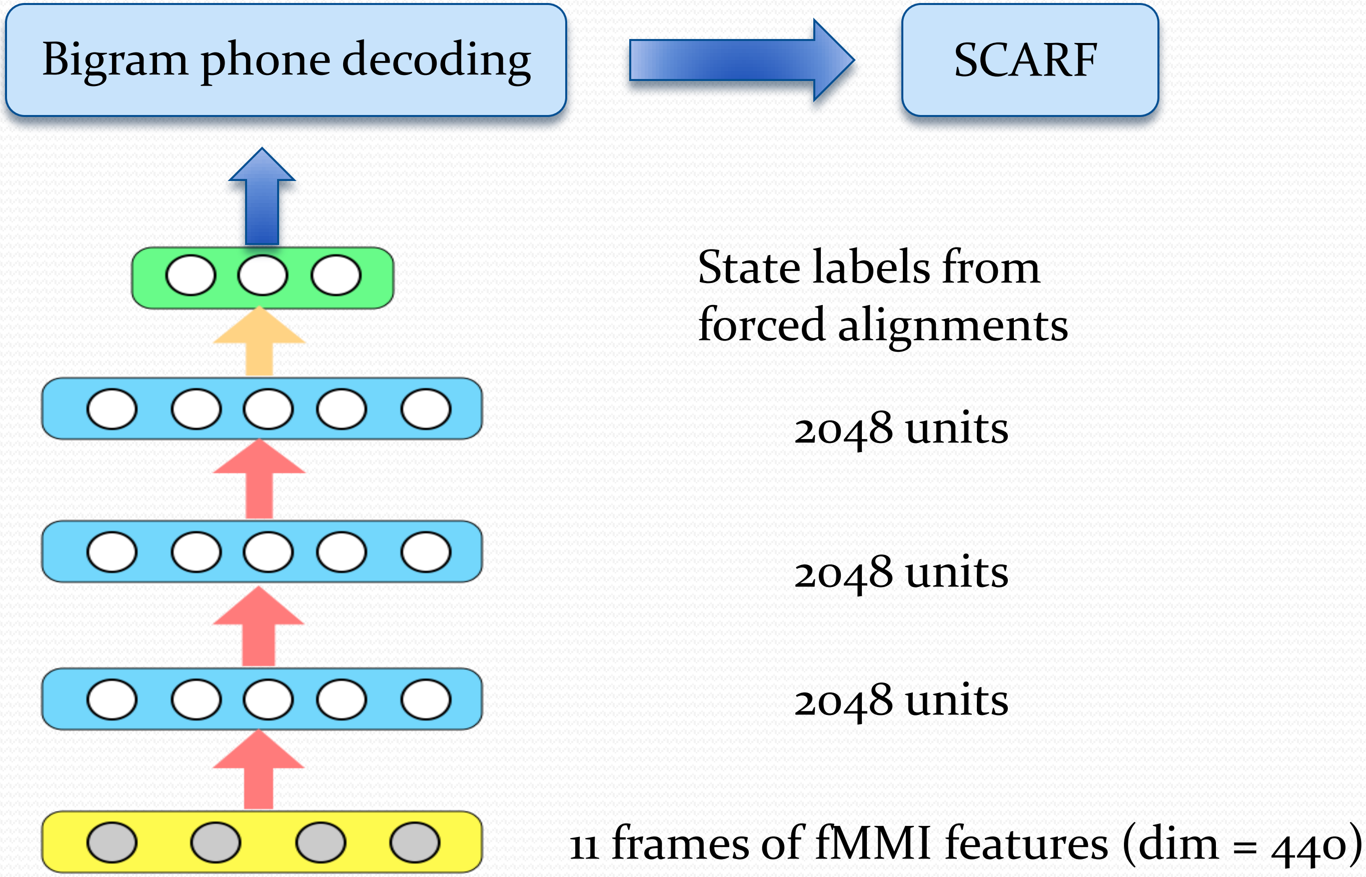
States represent whole words (not phonemes)

Log-linear model relates words to observations

Observations blocked into groups corresponding to words. Observations typically detection events.

Does ***ay*** appear in this segment?

# Current setup of deep nets



Bigram phone decoding

SCARF

State labels from
forced alignments

2048 units

2048 units

2048 units

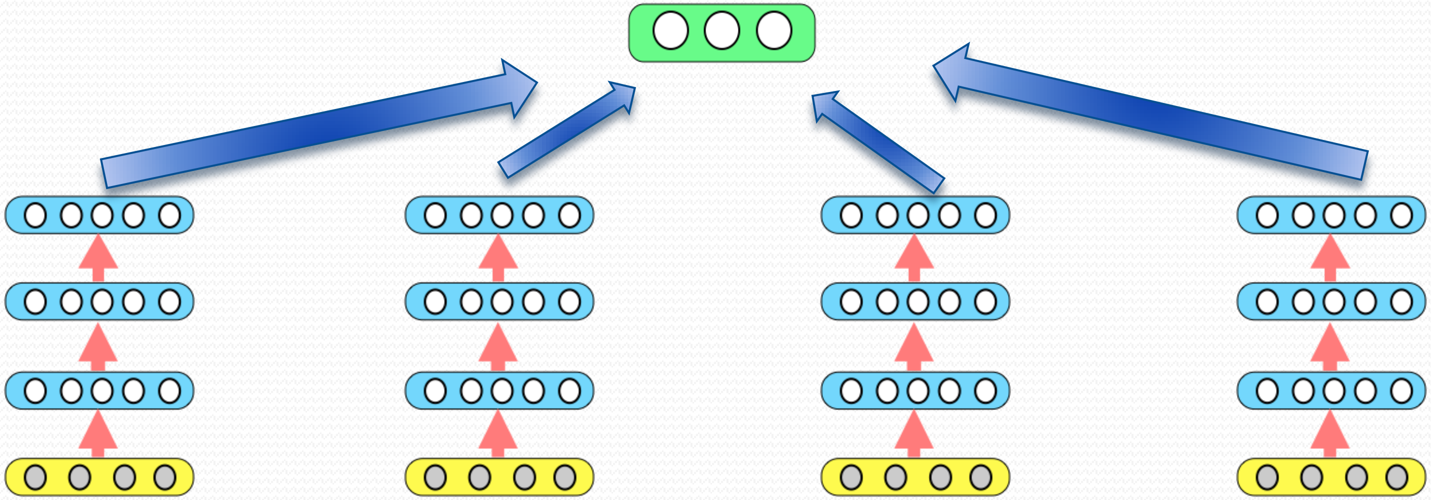
11 frames of fMMI features (dim = 440)

# Main accomplishments (a)

- Successful application to large-vocabulary speech recognition
  - Existing work is on TIMIT (3-hour data).
  - Our work is on Broadcast News (430-hour data).
- Improvement over state-of-the-art baseline systems
  - Use SCARF to integrate deep net results as well as other useful features and systems
  - Reduce WER from 15.3% to 15.1%

# Main accomplishments (b)

- Implementation of deep nets on clusters
  - Existing approach: sequential processing on single GPU
  - Our approach: parallel training on CPU clusters
  - Impact:  deep nets become mainstream



# Use deep nets as acoustic model

| Acoustic Information | Phone error rate | Word error rate |
|---|---|---|
| None | -- | 17.9 |
| **Deep Net 20hr *** | **28.8** | **17.1** |
| **Deep Net 40hr *** | **28.2** | **17.0** |
| FDLP-M 430hr | 28.9 | 16.9 |

Use much less data, but starts from fMMI features

More training data helps

1% absolute improvement

* fMMI input features trained on 430 hrs

# Integrating all detectors

| Acoustic Information | WER Trigram LM |
|---|---|
| SCARF1 + MSR | 15.3% |
| + FDLP-M | 15.1 |
| + Deep Net 20hr | 15.1 |
| + Deep Net 40hr | 15.2 |
| 8 Streams | 15.0 |

0.3% absolute improvement

## Take-home messages

Every detector improves a bit.

Integration improves too , but not additively.

Preliminary diagnosis

high correlations with baselines

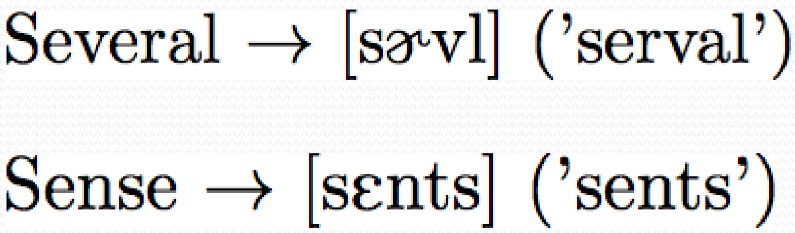
# TF-IDF and Pronunciation Variation

Sam Bowman

Geoffrey Zweig

# Pronunciation Variation

- There is no guarantee that speakers will produce the dictionary-form pronunciations of words…
- …nor is there a guarantee that our detectors will correctly identify the segments that they do produce.



Several → [sɚvl] (’serval’)

Sense → [sɛnts] (’sents’)

- I worked on two novel models that address that variation within SCARF.
  - Decision-Tree modeling & TF-IDF
  - Focusing on TF-IDF here (time constraints)

# TF-IDF in ASR

- The SCARF toolkit contains a TF-IDF–based decoder which models the correspondence between words and observed pronunciations, and can learn systematic variation.
- We borrow the Term Frequency–Inverse Document Frequency (TF-IDF) metric from the information retrieval community:
    - TF-IDF scores quantify the degree to which a phone n-gram is characteristic of the known pronunciations of a word.

# TF-IDF

- Intuitively, TF-IDF weights the frequency of n-gram (term) *j* in tokens of word (document) w against the overall frequency of *j* in all words (*W*).
- Adapted from Zweig, Nguyen, Droppo and Acero 2010: for the position corresponding to segment *j* in word *w*:

$$TFIDF(j, w) = \frac{n_j}{N_w} \log \frac{W}{d_j}$$

- These values are computed for every (*word, phone n-gram*) pair:

EITHER(01): AA : 0.6, AE : 0.1, AO : 0.0, AY : 2.2...

(unigrams are used here for simplicity)

# TF-IDF

- When hypotheses are scored, the hypothesis is converted to an analogous vector, and the two vectors are compared by the cosine similarity heuristic:

$$\frac{V_w \cdot V_{hyp}}{|V_w||V_{hyp}|}$$

- N-grams indirectly but effectively capture the ordering of sub-word units within the words.
- This produces a score from 0 (no match) to 1 (perfect).
- We can use these scores in a freestanding recognizer, or to annotate existing lattices.

# The Dictionary

- Our TF-IDF vectors are derived from observed pronunciations.
- Our most successful dictionary incorporates canonical pronunciations from a conventional dictionary and observed pronunciation variants from training data.

| | | | |
|---|---|---|---|
| EITHER | AY DH ER | 12 | AY DH AH |
| EITHER | AY DH ER | 203 | AY DH ER |
| EITHER | AY DH ER | 2 | AY TH AO T |
| EITHER | AY DH ER | 2 | AY V |
| … | | | |
| EITHER | IY DH ER | 2 | IH Z |
| EITHER | IY DH ER | 486 | IY DH ER |
| EITHER | IY DH ER | 2 | L IY D ER |
| … | | | |

# TF-IDF: WER Results

| | WER |
|---|---|
| Direct Recognition | 22.9% |
| SCARF1 + MSR | 15.3 |
| + TF-IDF | 15.2 |

- Direct recognition from detections possible with TF-IDF!
- Some improvement from using TF-IDF scores  as additional information

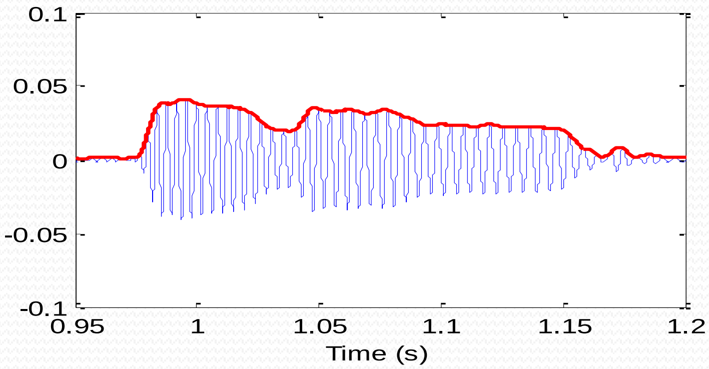
# Modulator-based Acoustic Features for Speech Recognition

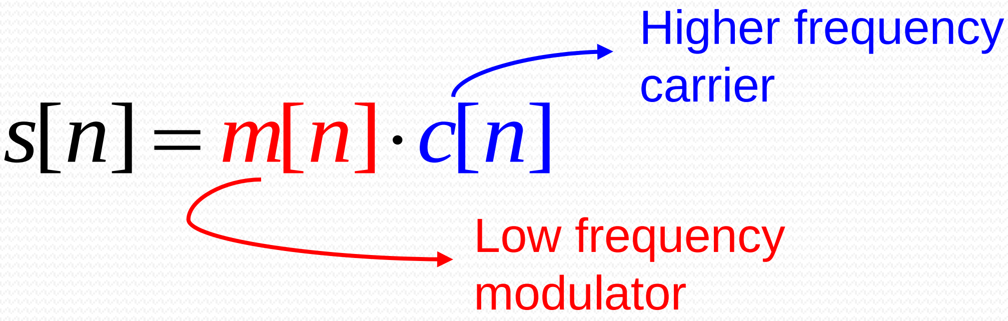
Pascal Clark
U Washington

Greg Sell
Stanford

Les Atlas
U Washington

# Why use modulators?





$$s[n] = m[n] \cdot c[n]$$

- Modulators capture salient long-term speech components (2 – 50 Hz syllabic and phonetic rates)

- Modulators are bandlimited and robust to carrier interference (e.g., pitch)

- Modulators can provide new and complementary information for speech recognition via SCARF

# How to find modulators

- Two novel, complementary approaches
  - ***Convex Demodulation***
  - ***Coherent Demodulation***

- Both approaches start with a sum-of-products model:

$$s[n] = \sum_k s_k[n] = \sum_k m_k[n] \cdot c_k[n]$$

Speech signal

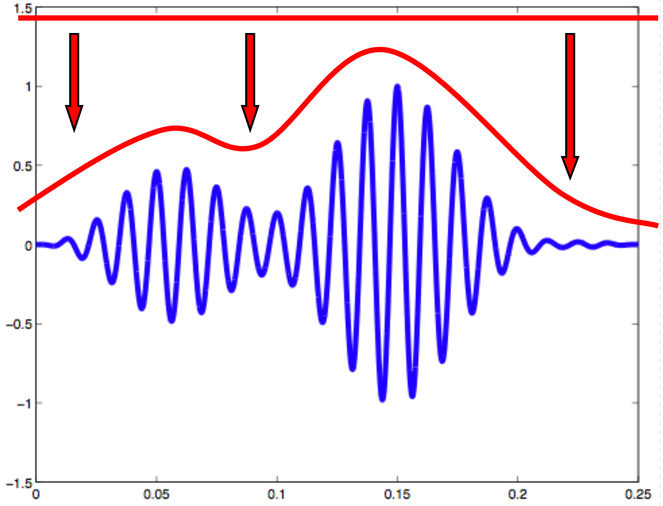
Subband signals

Modulators

Carriers

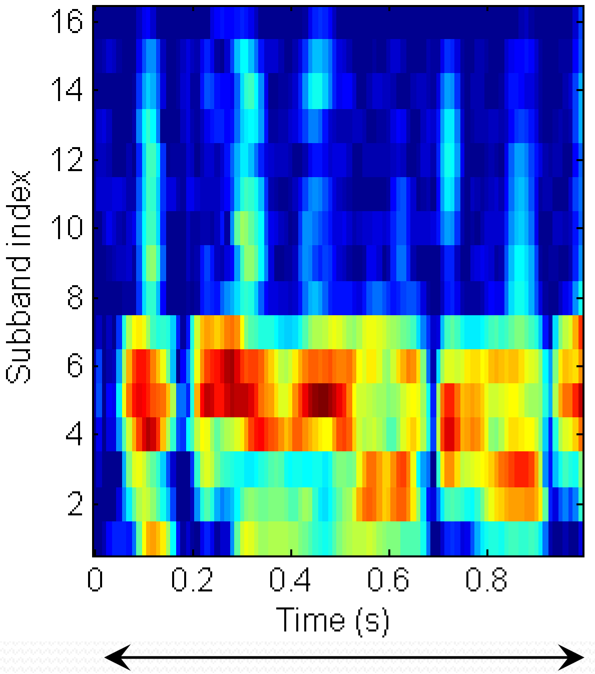
# Method 1 - Convex Demodulation

**Each subband:**



**Modulator is constrained by local maxima and smoothness, and stored in an array:**

***Convex modulator feature array:***



Bandlimited across frames

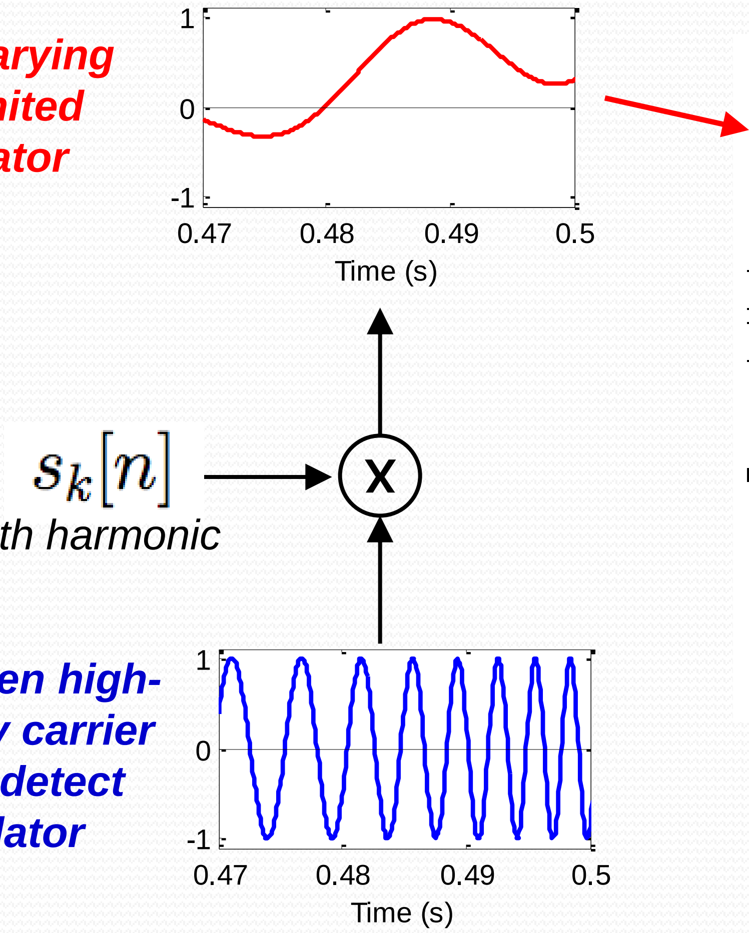
**Carrier fine structure is discarded so SCARF sees speech information from the modulators *only***

***Accomplishment:***

*Training Samuel's MLP phoneme recognizer on these features led to 0.4% word-error rate improvement using a trigram language model in SCARF.*

# Method 2 – Coherent Demodulation

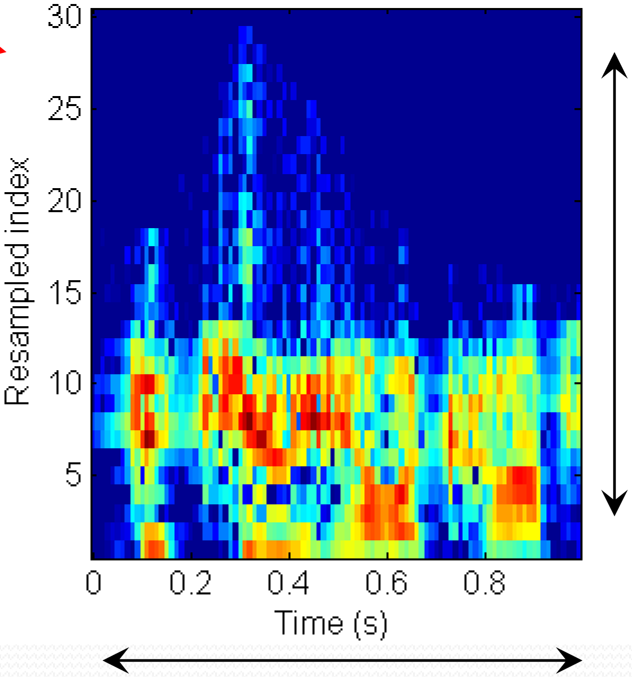
***Slowly-varying bandlimited modulator***



*k-th harmonic*

***Pitch-driven high-frequency carrier used to detect modulator***

***Coherent modulator feature array:***



Pitch-invariant representation (after cross-channel resampling)

Bandlimited *across* frames

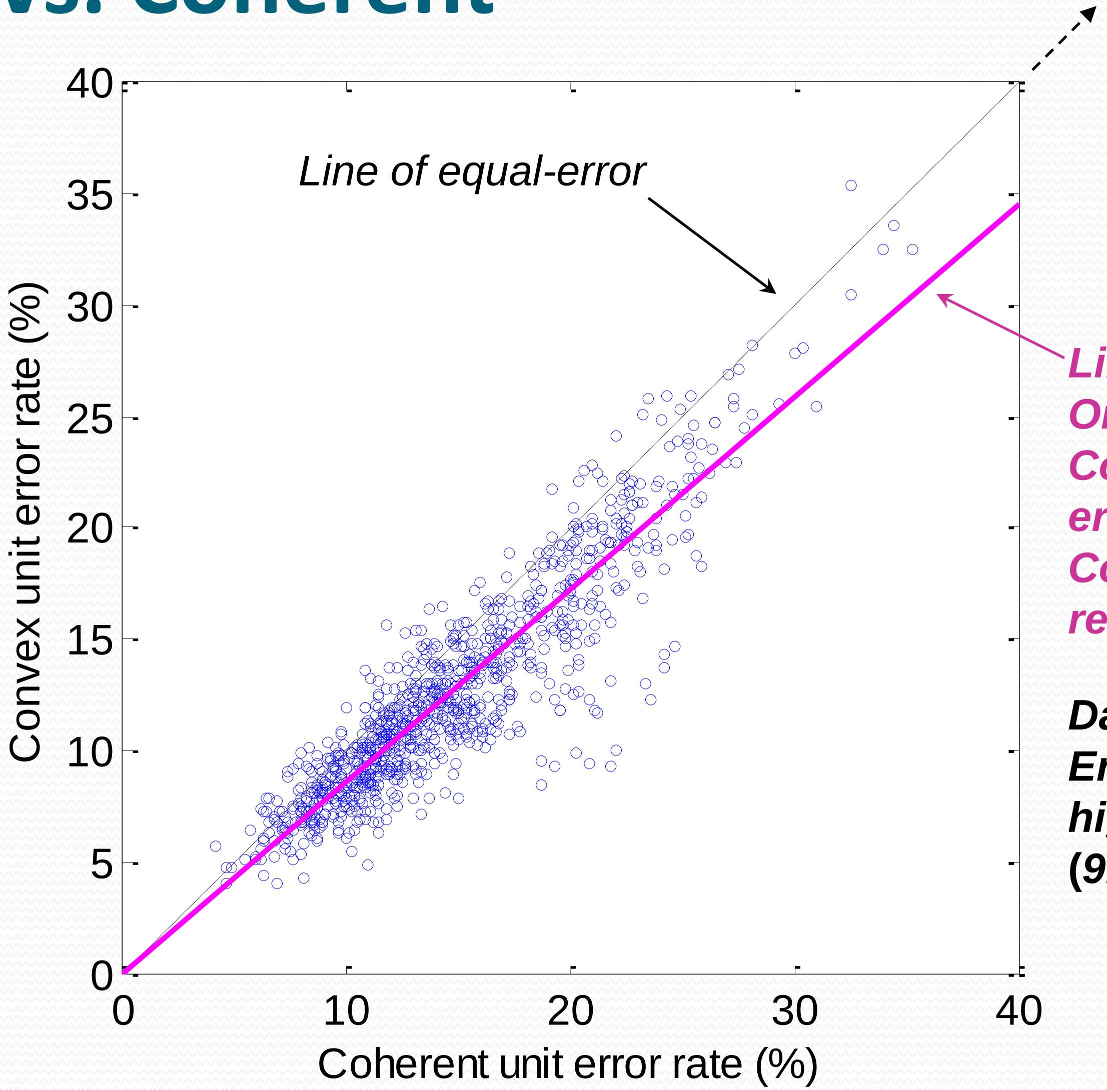
For more demos: **http://isdl.ee.washington.edu/projects/modulationtoolbox/**

# Method Comparison

| | **Convex** | **Coherent** | **Conventional (Hilbert, full/half-wave rectification** |
|---|---|---|---|
| **Bandlimited m[n] and c[n]?** | **Yes** | **Yes** | **No** |
| Modulator Constraints | Non-negative, Real | None | Non-negative, Real |
| Carrier Constraints | None | Complex, Narrowband | None |

# Max. entropy-based unit classification error: Convex vs. Coherent

*(Chance is 50%)*

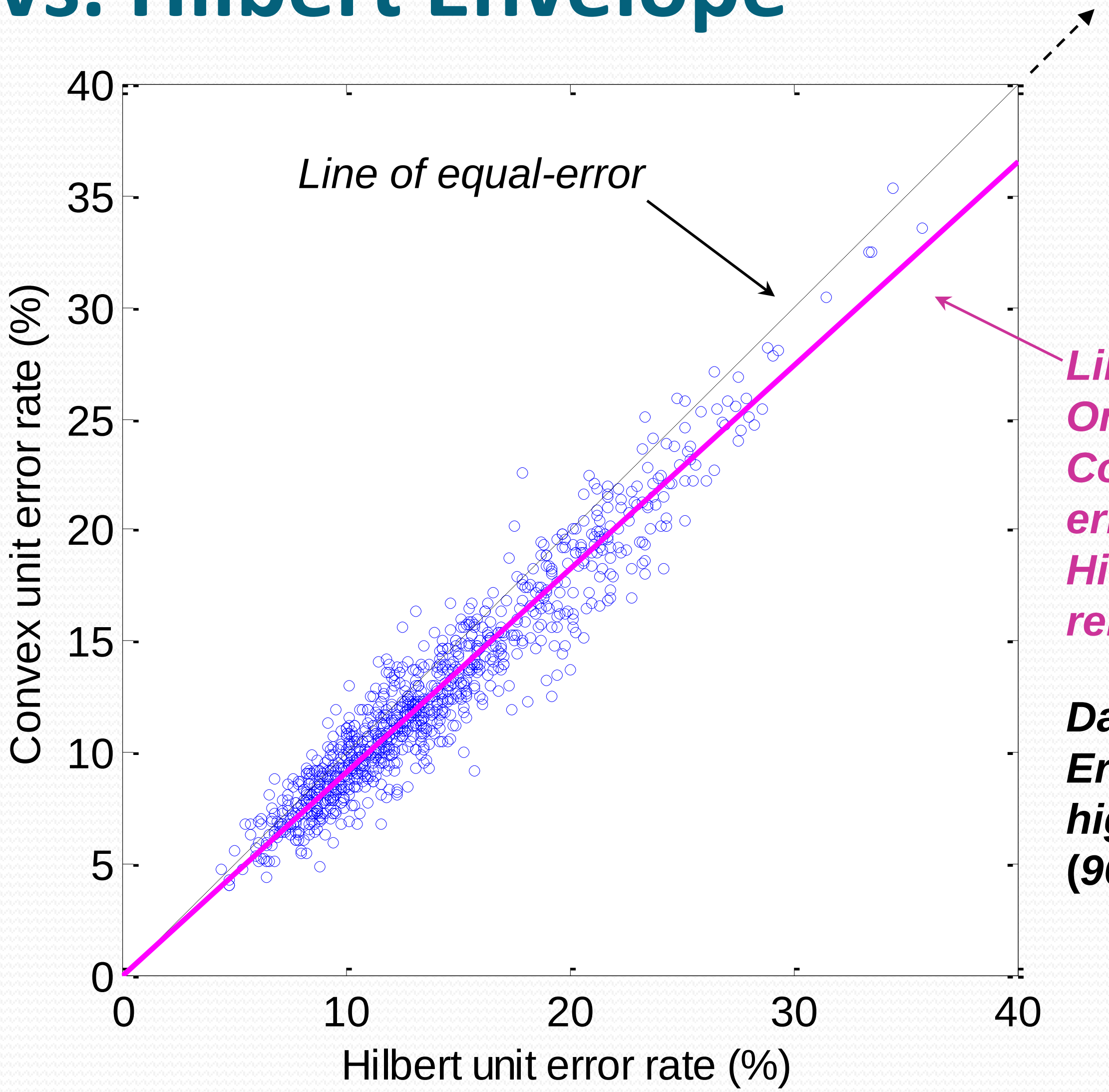


***Linear regression: On average, Convex is less error-prone than Coherent (86% relative error rate)***

***Data spread: Error rates are highly correlated (91%)***

# Max. entropy-based unit classification error: Convex vs. Hilbert Envelope

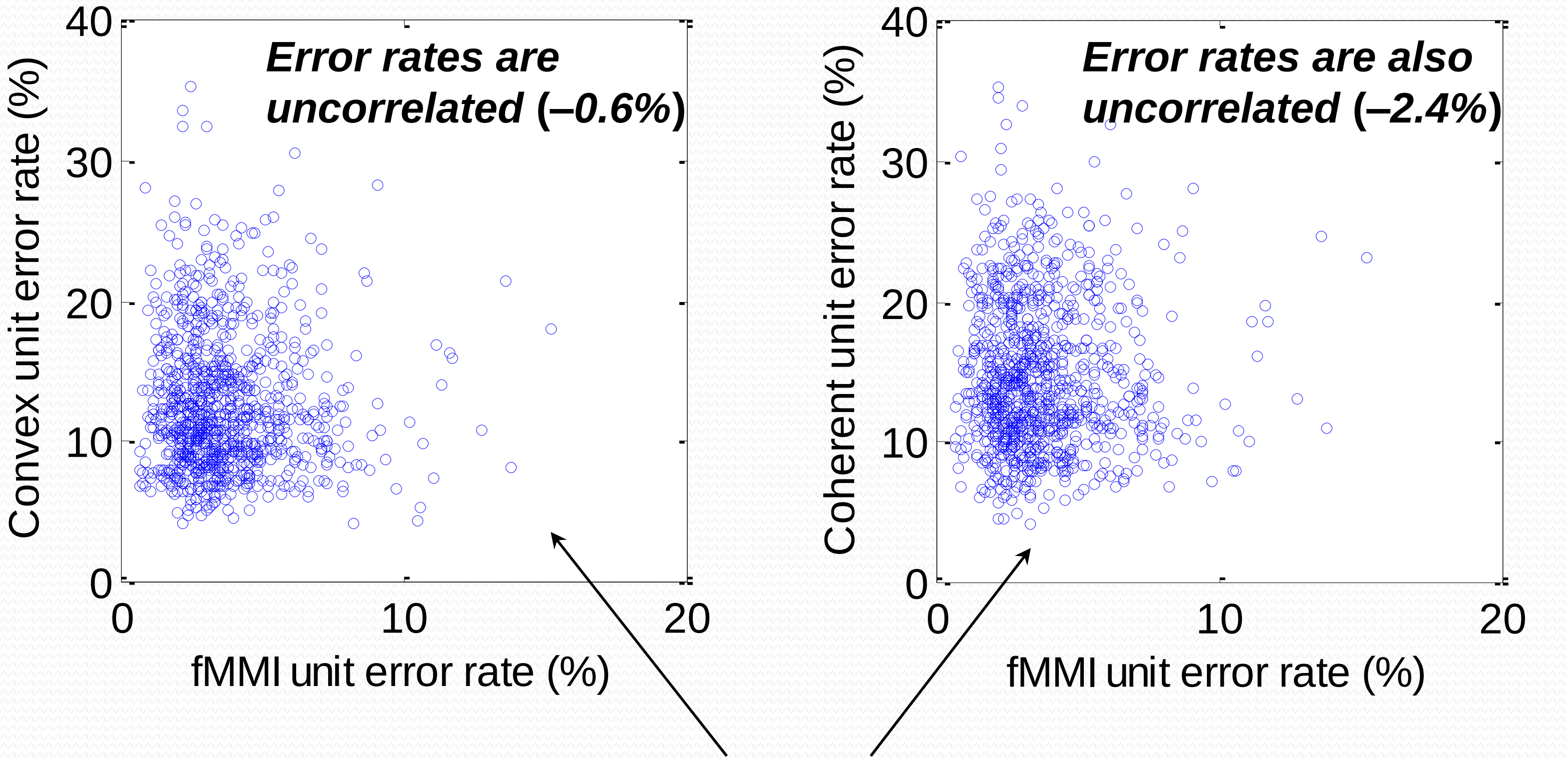
*(Chance is 50%)*



***Linear regression: On average, Convex is less error-prone than Hilbert (91% relative error rate)***

***Data spread: Error rates are highly correlated (96%)***

# Max. entropy-based unit classification error: Convex, Coherent vs. fMMI features



***Error rates are uncorrelated (–0.6%)***

***Error rates are also uncorrelated (–2.4%)***

*Highly complementary to the baseline features: possible new viewpoints to add to SCARF*

# Comparison to Standard Features

| **Standard Features** | | **Modulation Features** |
|---|---|---|
| *Mean Normalization* | | *Mean Normalization* |
| MFCC | Mean subtraction | Applicable |
| *Speaker Adaptation* | | *Speaker Adaptation* |
| VTLN | Spectral warping | Spectral resampling |
| fMLLR | Move features toward phoneme Gaussians | Applicable |
| *Discriminative Transforms* | | *Discriminative Transforms* |
| HLDA | Dimensional reduction | Applicable |
| fMMI | Region-dependent feature offsets | Applicable |

# Summary

- We introduced two bandlimited modulation signal models for speech recognition: **Coherent** and **Convex**
- Convex shows a preliminary improvement over conventional Hilbert envelopes
  - Potential for further development as a new bandlimited foundation for MFCCs and fMMI features
- Both Coherent and Convex are highly complementary to the baseline features in a speech classification task

# Modeling Duration as an External Feature for SCARF:

## The Discriminative Ability of Word and Phone Durations

Justine Kao

Patrick Nguyen

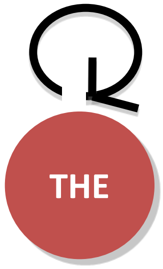
# Outline

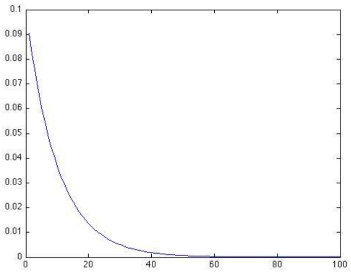
- Introduction to duration modeling
- Duration features
  - Probability density function features
  - Phone duration features
  - Word span confusion features
  - Log probability density function features
  - Discretized (bucketed) features
  - Pre- and post-pausal features
- Summary of results
- Discussion and Questions

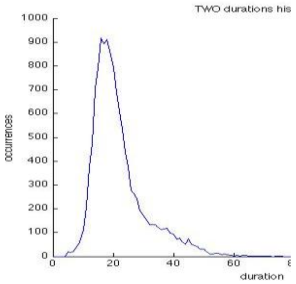
# Duration models

## Motivation and background

- **What is a good feature?**
  - Something that measures the consistency between a word hypothesis and the underlying acoustics
- **Duration features**
  - Word duration should  be able to provide information about word identity
  - HMM
    - Duration of a state (word, phone, etc) modeled as probability of remaining in that state → exponential model
    - Difficult to model true duration distributions







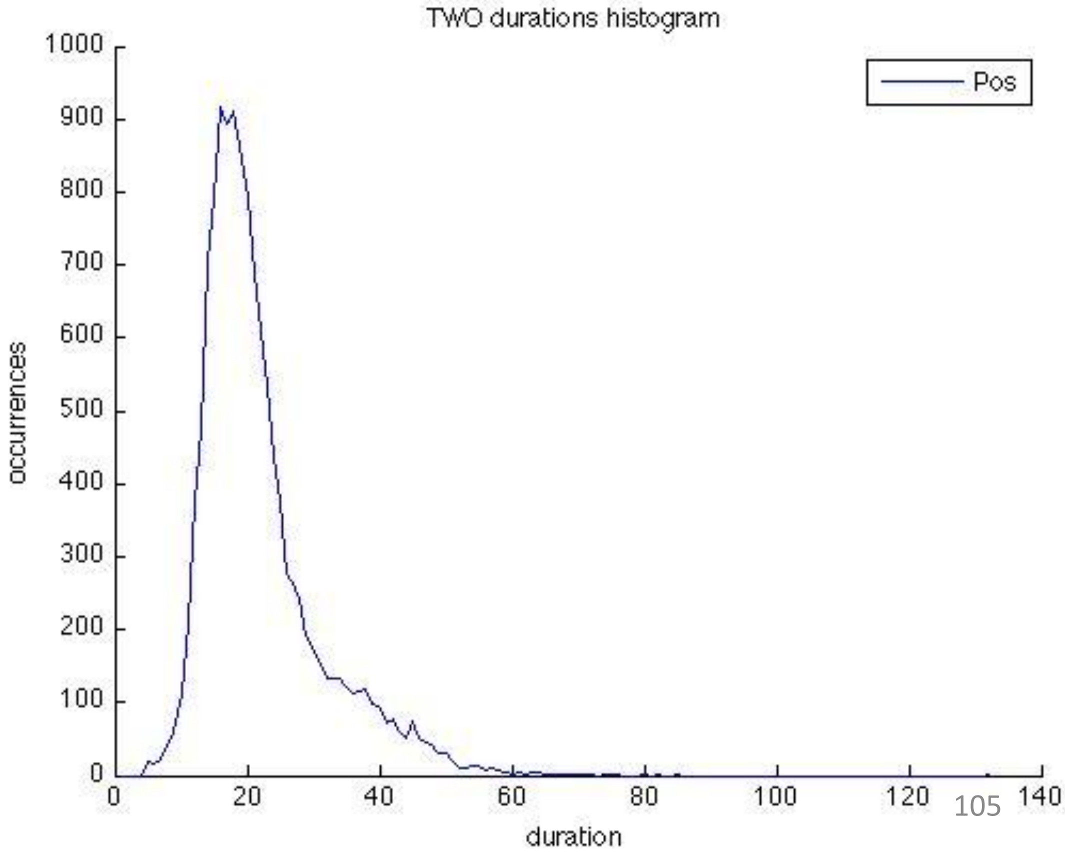
# Duration models ●

## Discriminative Duration Models

- If there are differences between the duration distributions of correct and incorrect word hypotheses, then word duration could be a useful feature to discriminate between them
- Model this difference to come up with duration features

- **Are they different?**
  - Find all hypotheses of "TWO" that are **correct**
  - → positive examples

  - Plot their durations

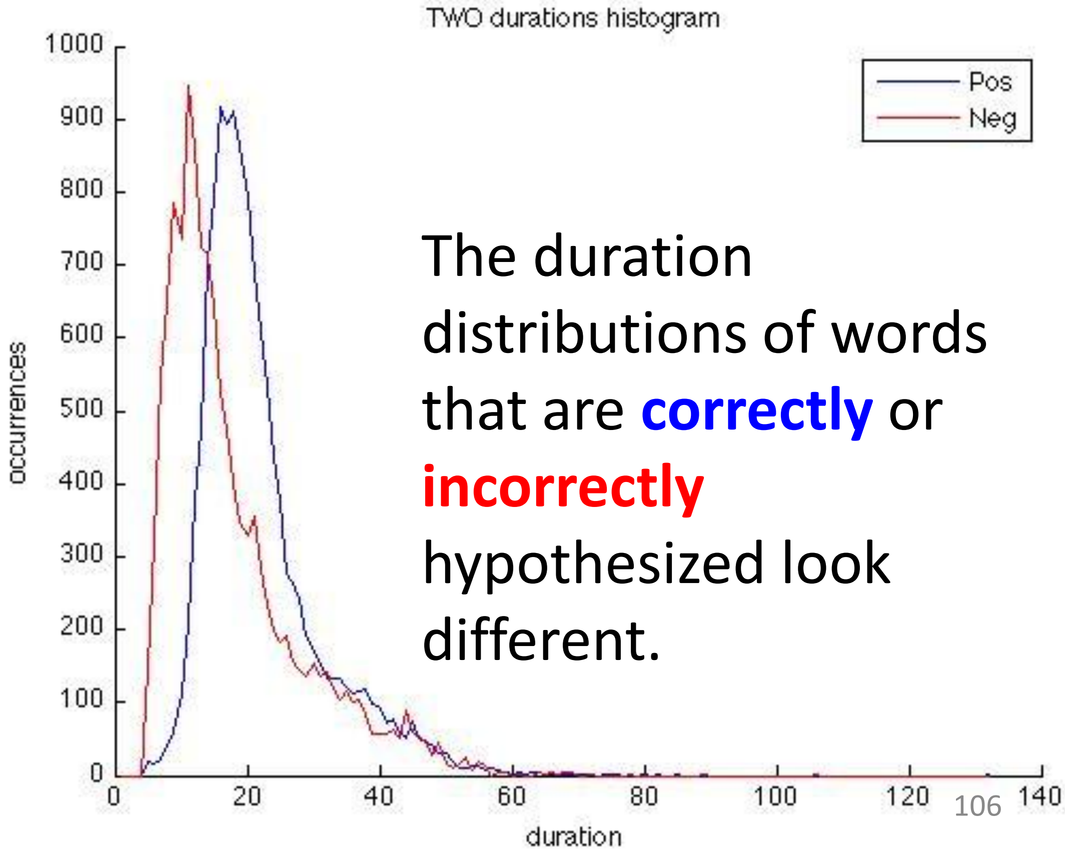


# Duration models

## Discriminative Duration Models

- If there are differences between the duration distributions of correct and incorrect word hypotheses, then word duration could be a useful feature to discriminate between them
- Model this difference to come up with duration features

- **Are they different?**
  - Find all hypotheses of “TWO” that are **incorrect**
  - → negative examples
  - Plot their durations



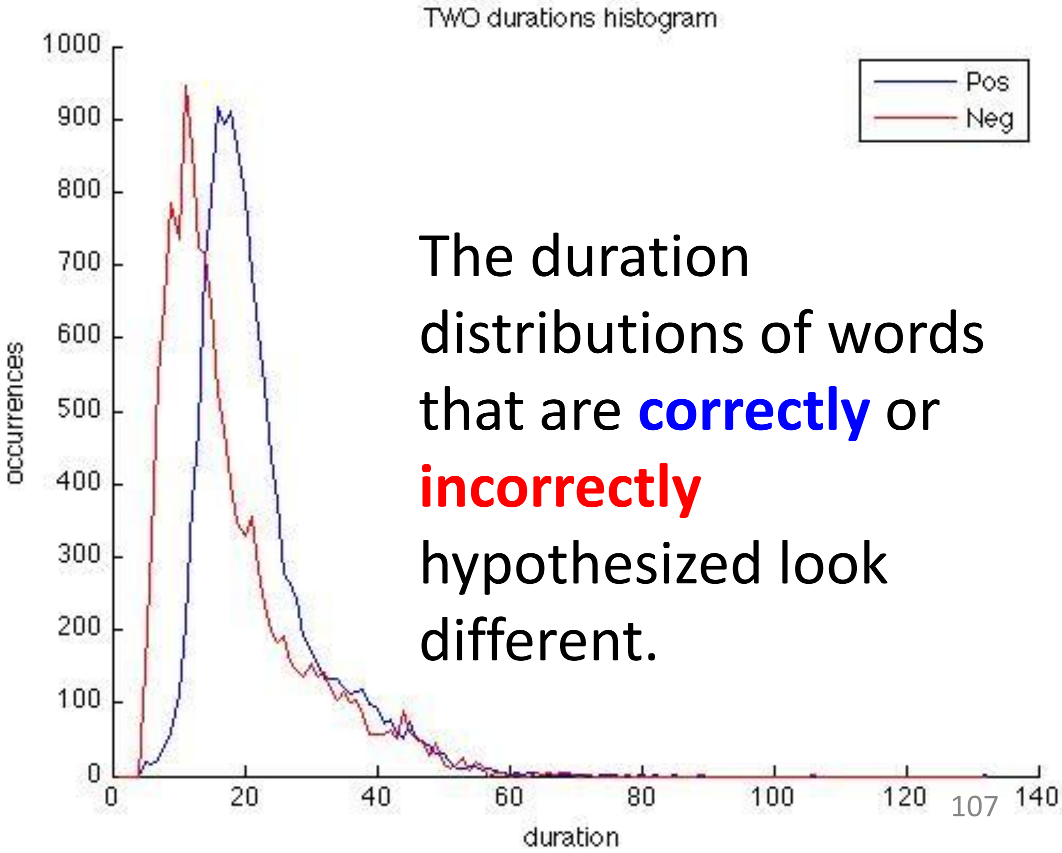
The duration distributions of words that are **correctly** or **incorrectly** hypothesized look different.

# Duration models ●

## Discriminative Duration Models

- If there are differences between the duration distributions of correct and incorrect word hypotheses, then word duration could be a useful feature to discriminate between them
- **Model this difference to come up with duration features**

- **Are they different?**
  - Find all hypotheses of "TWO" that are **incorrect**
  - → negative examples
  - Plot their durations



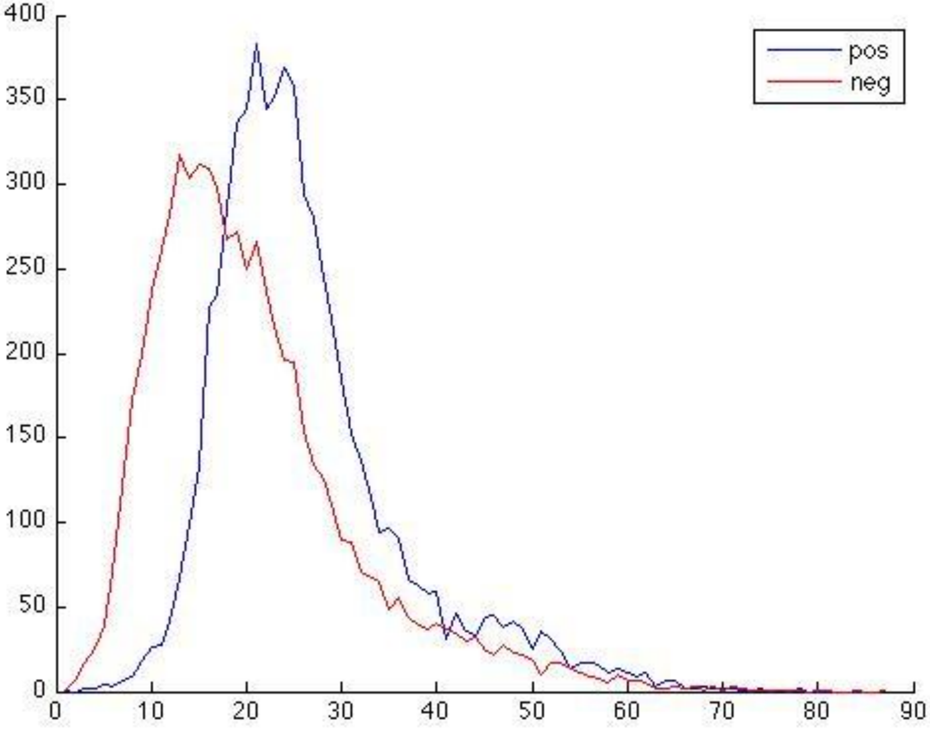
The duration distributions of words that are **correctly** or **incorrectly** hypothesized look different.

# Duration models

## Investigating word durations

- Focus on the top 100 most frequent words seen in the training transcripts

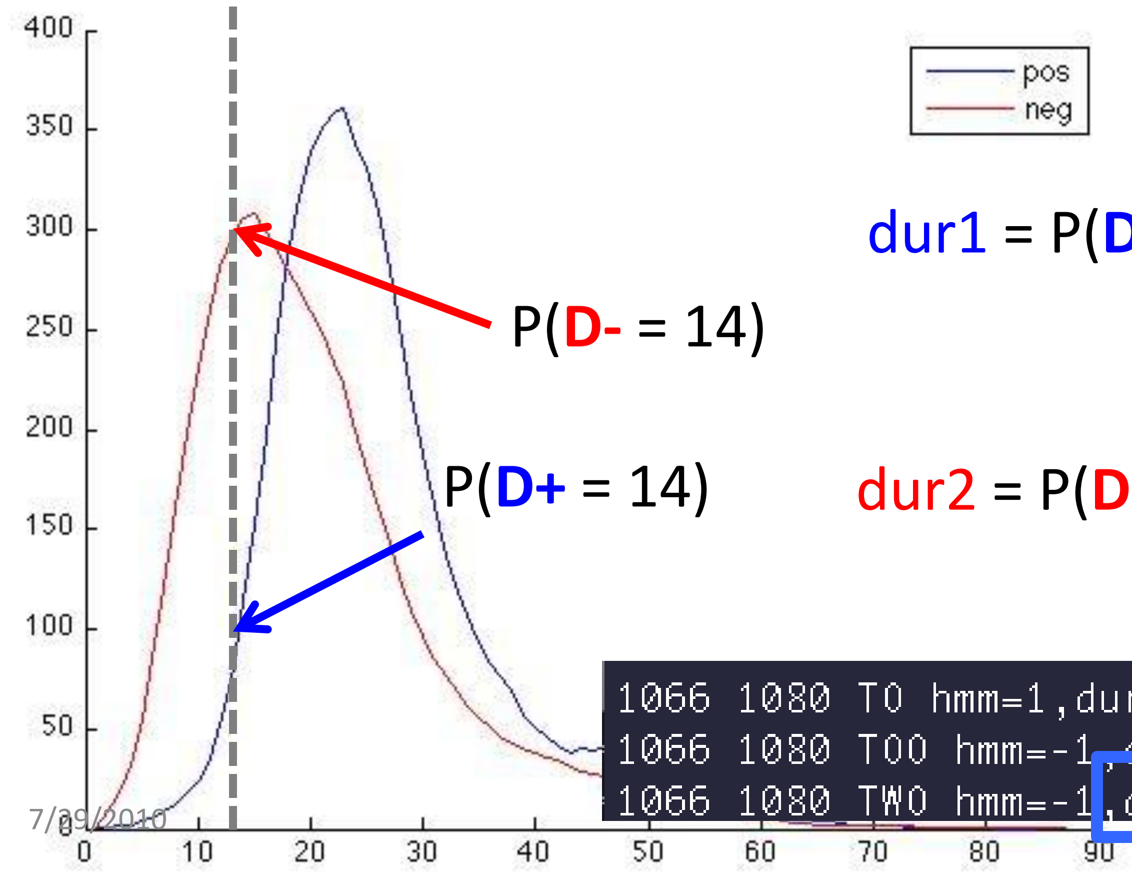
- **Large portion of data:** The top 100 most frequent words account for 47.5% of all word occurrences in the training set transcript
- **Large portion of important data:** The top 100 most frequent words account for 48.58% of all errors in the test set
    - Function words, shorter



# Duration as feature

## Probability density function scores for high-frequency words

- **Task:** given the duration of a word hypothesis, capture the likelihood of it being in the correct or incorrect distribution
  - Suppose a word hypothesis “TWO” is 14 frames long



P(D- = 14)

P(D+ = 14)

$$\text{dur1} = P(D+ = 14) = \frac{\text{num } (D+ = 14)}{\text{num } (\text{“TWO”} +)}$$

$$\text{dur2} = P(D- = 14) = \frac{\text{num } (D- = 14)}{\text{num } (\text{“TWO”} -)}$$

```
1066  1080  TO  hmm=1,dur1=0.092314,dur2=0.077221
1066  1080  TOO  hmm=-1,dur1=0,dur2=0
1066  1080  TWO  hmm=-1,dur1=0.005578 dur2=0.040293
```

# Duration as feature ○

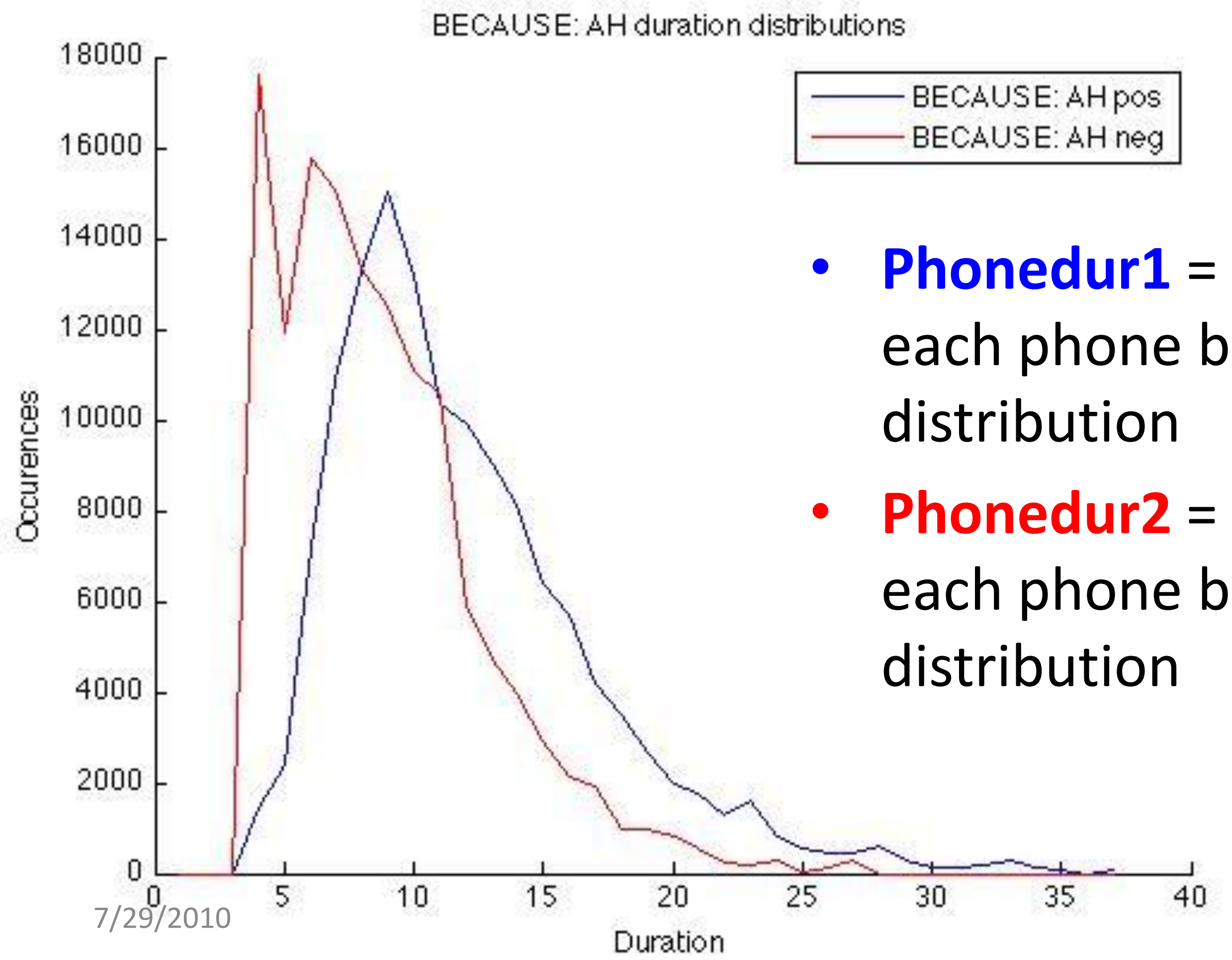
## Results: take 1

| No. | System | Dev |
|---|---|---|
| t.0 | SCARF1 + MSR | 15.3% |
| t.1 | t.0 + word duration scores | 15.2% |
| | | |
| | | |

# Phone durations as feature

## Probability density functions for each phone in a word

- Phones of correctly and incorrectly hypothesized words also have different duration distributions



- **Phonedur1** = sum of log likelihood of each phone being in a positive distribution
- **Phonedur2** = sum of log likelihood of each phone being in a negative distribution

# Phone durations as feature

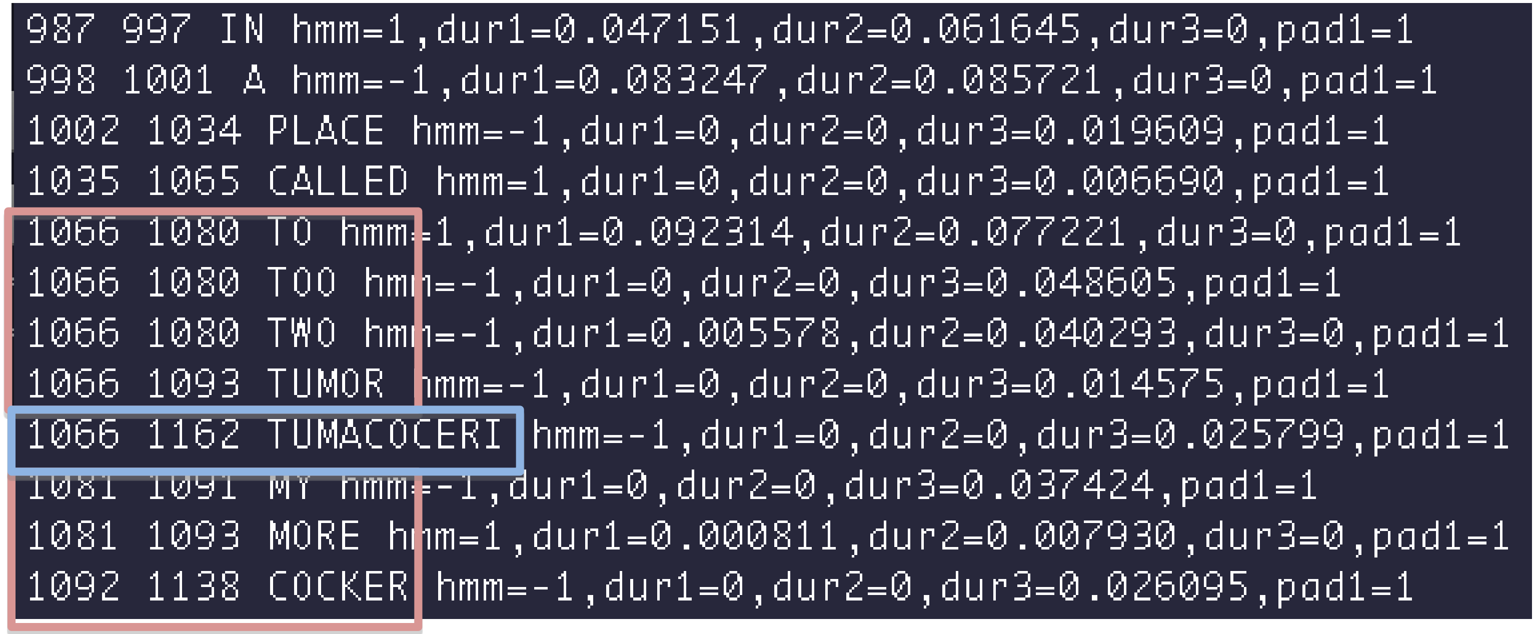
## Results: take 2

| No. | System | Dev |
|---|---|---|
| t.0 | SCARF1 + MSR | 15.3% |
| **t.1** | **t.0 + word duration scores** | **15.2%** |
| **t.2** | **t.1 + phone duration scores** | **15.1%** |
| | | |

# Duration as feature

## Word span confusions

- Long words are sometimes confused with a sequence of shorter, more high-frequency words



```
987 997 IN hmm=1,dur1=0.047151,dur2=0.061645,dur3=0,pad1=1
998 1001 A hmm=-1,dur1=0.083247,dur2=0.085721,dur3=0,pad1=1
1002 1034 PLACE hmm=-1,dur1=0,dur2=0,dur3=0.019609,pad1=1
1035 1065 CALLED hmm=1,dur1=0,dur2=0,dur3=0.006690,pad1=1
1066 1080 TO hmm=1,dur1=0.092314,dur2=0.077221,dur3=0,pad1=1
1066 1080 TOO hmm=-1,dur1=0,dur2=0,dur3=0.048605,pad1=1
1066 1080 TWO hmm=-1,dur1=0.005578,dur2=0.040293,dur3=0,pad1=1
1066 1093 TUMOR hmm=-1,dur1=0,dur2=0,dur3=0.014575,pad1=1
1066 1162 TUMACOCERI hmm=-1,dur1=0,dur2=0,dur3=0.025799,pad1=1
1081 1091 MY hmm=-1,dur1=0,dur2=0,dur3=0.037424,pad1=1
1081 1093 MORE hmm=1,dur1=0.000811,dur2=0.007930,dur3=0,pad1=1
1092 1138 COCKER hmm=-1,dur1=0,dur2=0,dur3=0.026095,pad1=1
```

- Find word hypotheses confused with longer-span or shorter-span hypotheses
- System should learn to penalize low scores in these categories more heavily than hypotheses with no word span confusions

# Duration as feature

## Results: take 3

| No. | System | Dev |
|---|---|---|
| t.0 | SCARF1 + MSR | 15.3% |
| **t.1** | **t.0 + word duration scores** | **15.2%** |
| t.2 | t.1 + phone duration scores | 15.1% |
| **t.3** | **t.1 + word span confusion scores** | **15.0%** |

# Summary

## Main accomplishments

| No. | System | Dev |
|---|---|---|
| **t.0** | **SCARF1 + MSR** | **15.3%** |
| t.1 | t.0 + word duration scores | 15.2% |
| t.2 | t.1 + phone duration scores | 15.1% |
| **t.3** | **t.1 + word span confusion scores** | **15.0%** |

- Up to 0.3 % gain on 15.3% WER (SCARF1 + MSR system)
- Word and phone durations can help SCARF discriminate between correct and incorrect word hypotheses
- Word durations may help resolve confusion between competing hypotheses

**Thank you!**

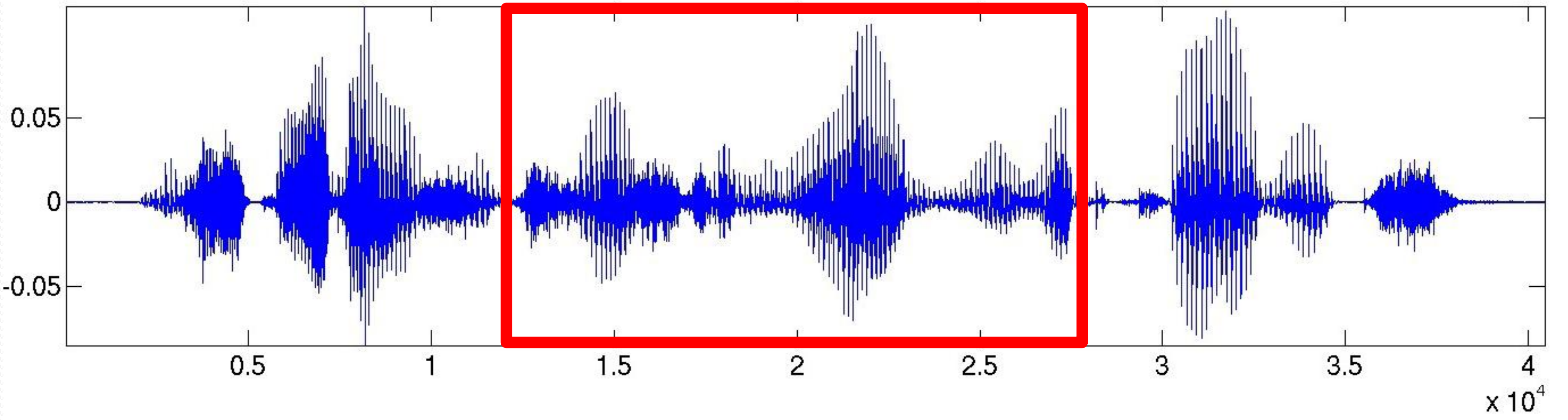
# Window-Based Syllable and Word Detectors

Geoffrey Zweig

Aren Jansen

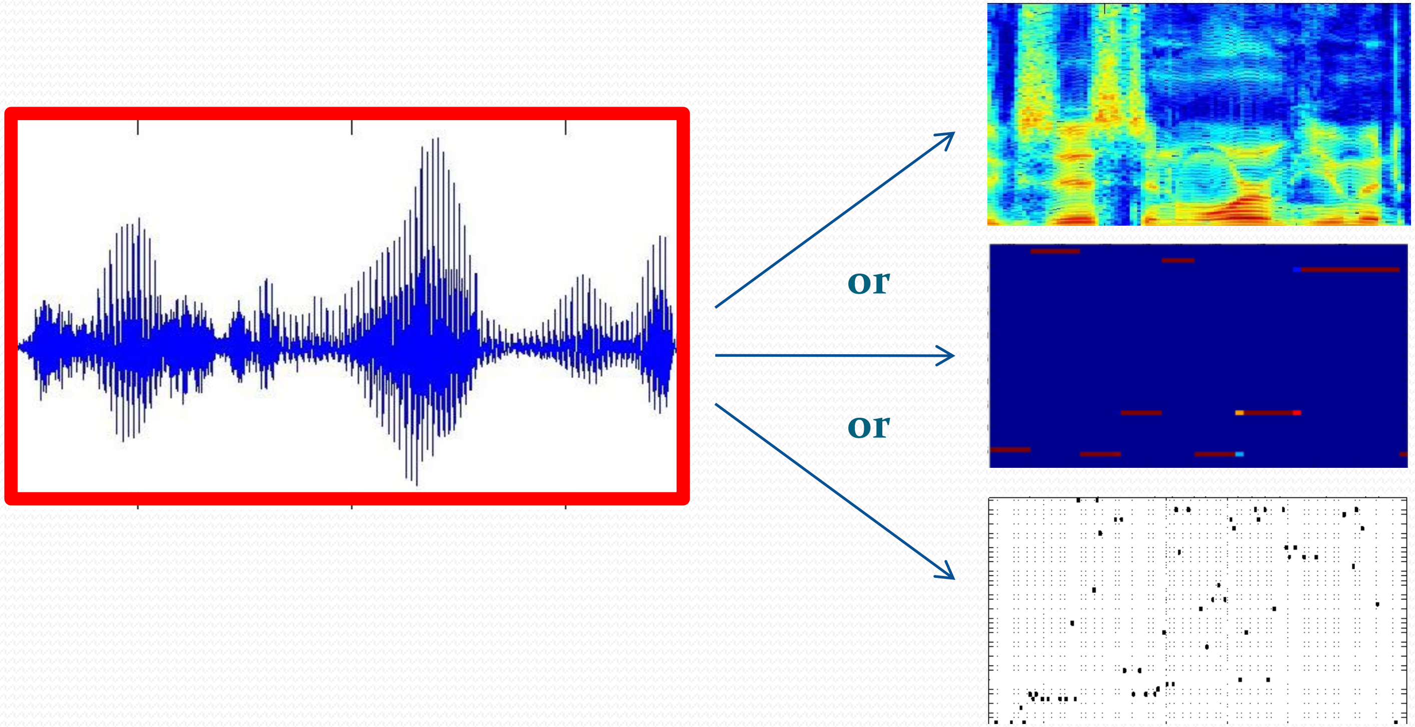
Keith Kintzley

# Window-Based Acoustic Models



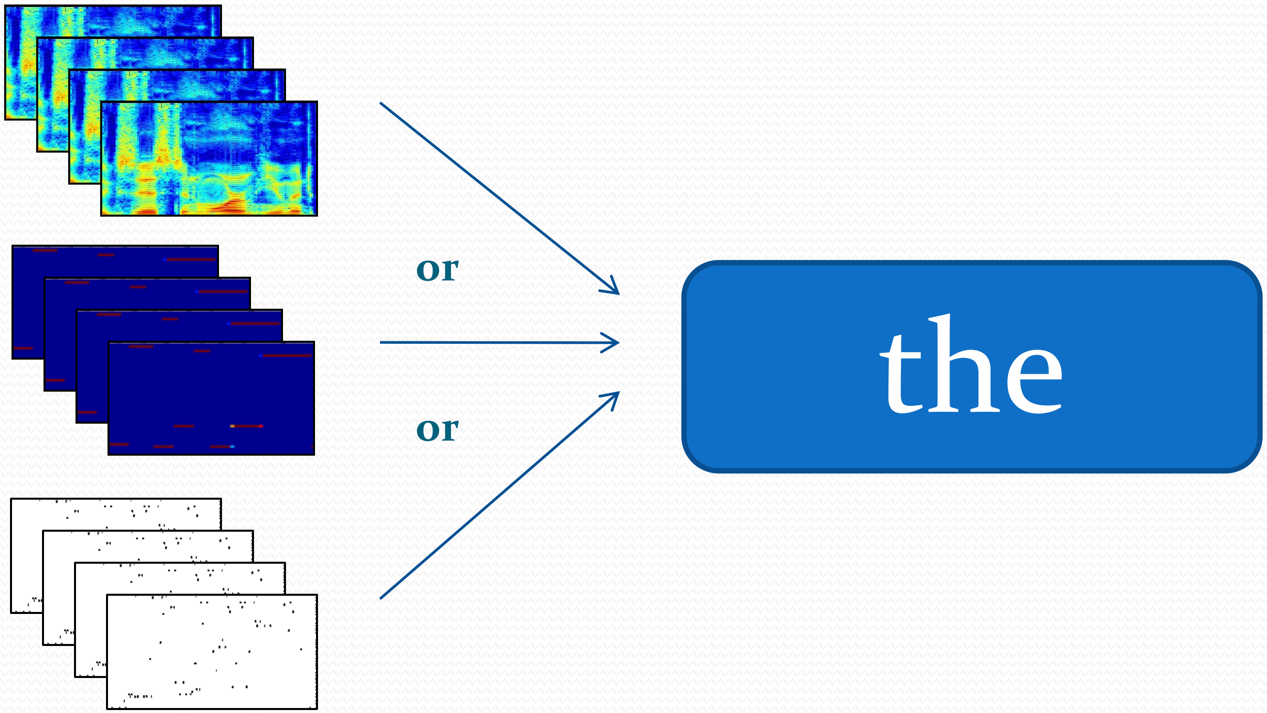
1. Collect examples of each unit (words, syllables, multi-phone units [MPUs])

# Window-Based Acoustic Models



2. Compute some representation for each example

# Window-Based Acoustic Models



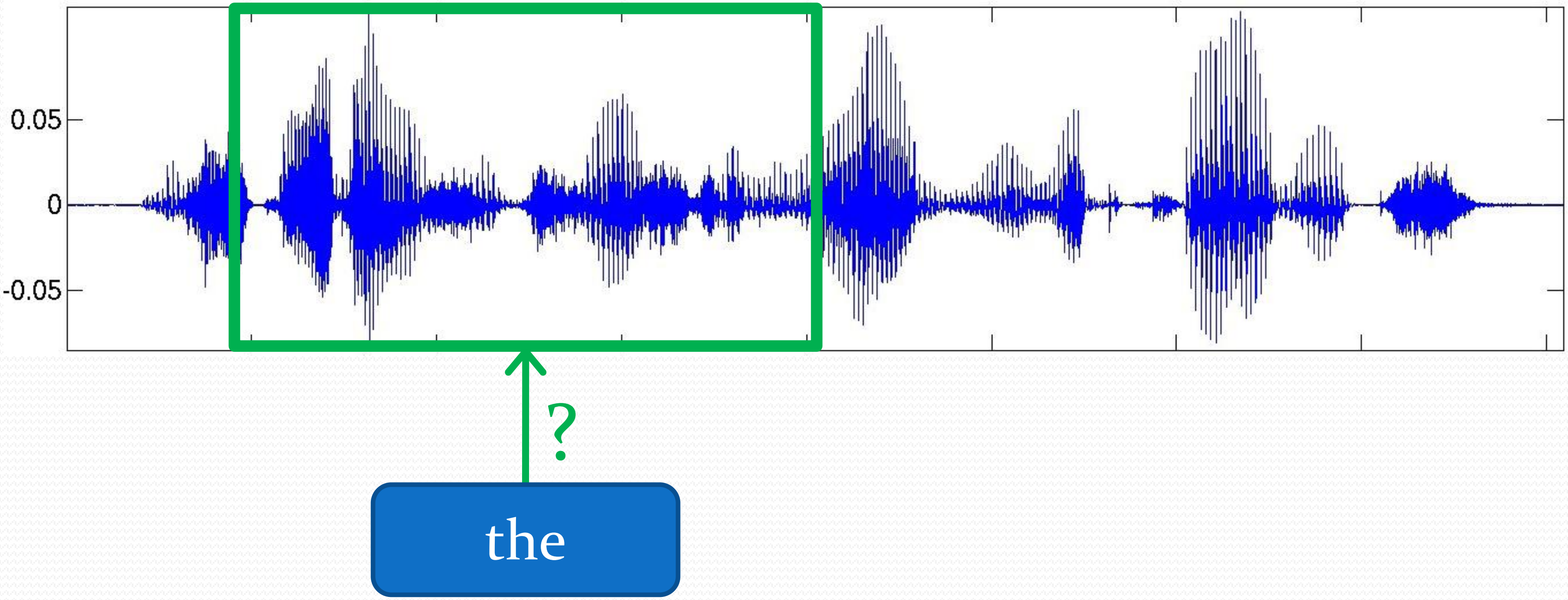
or

or

the

3. Build a model/classifier for each unit

# Window-Based Acoustic Models



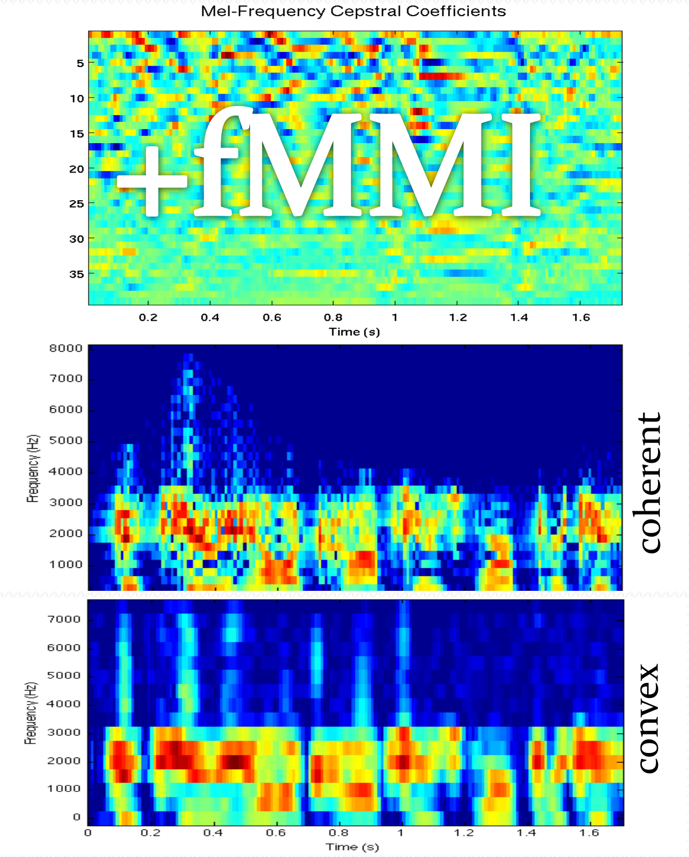
?

the

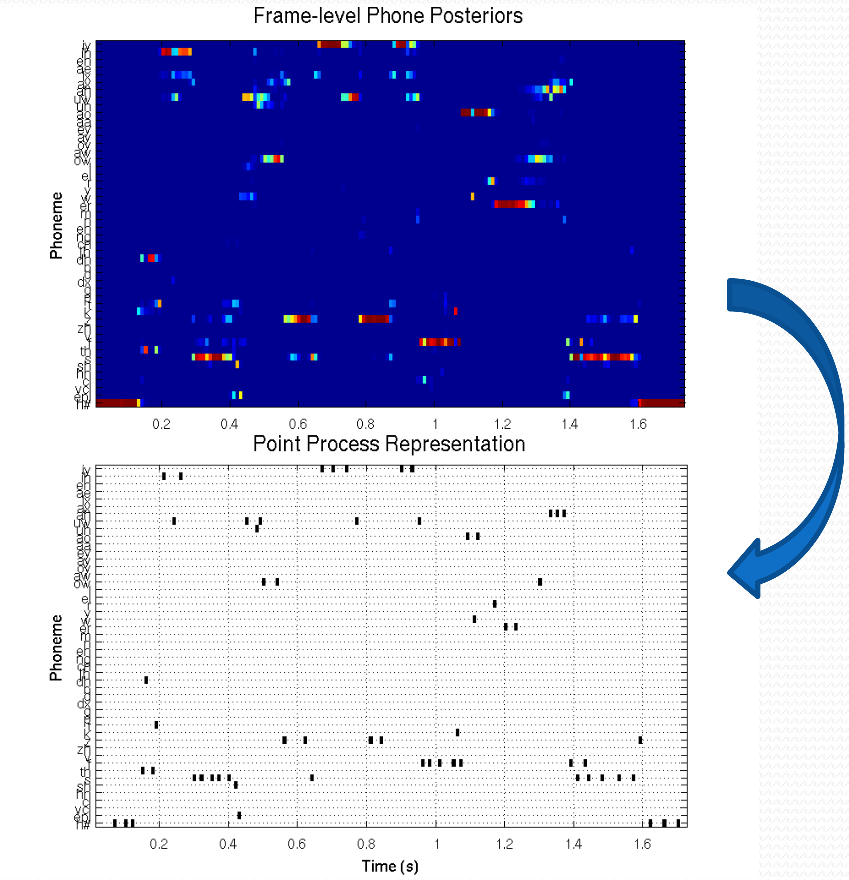
4. Detect or classify units in presented windows

# Representations

## Acoustic Feature Vectors



## MLP-Based Phonetic Events



# Window Types

## Fixed Windows

- Extract training examples with fixed sized windows (per unit)

**Benefit:**

Admits fixed-dimension vector space unit representation

**Drawback:**

No compensation for speaking rate variability

## Elastic Windows

- Normalize all examples to unit duration

**Benefit:**

Allows modeling of unit as a whole, regardless of duration

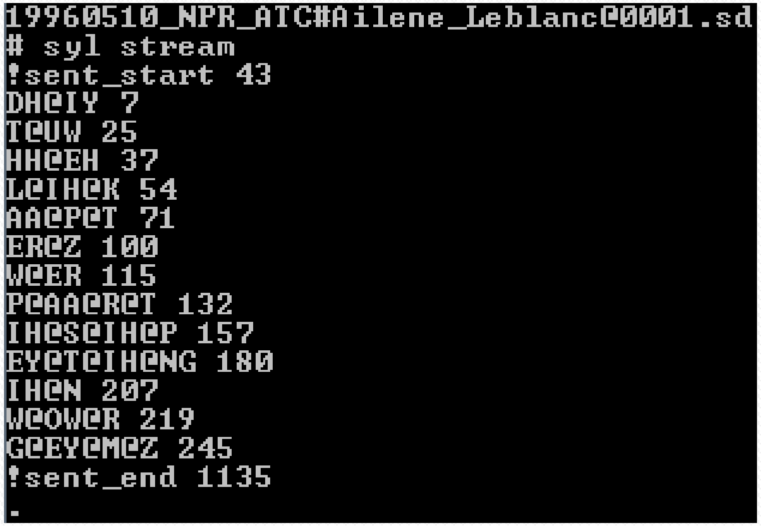
**Drawback:**

Normalization is difficult to get right, esp. with frames

# SCARF Integration Modes

## MPU Detector Streams

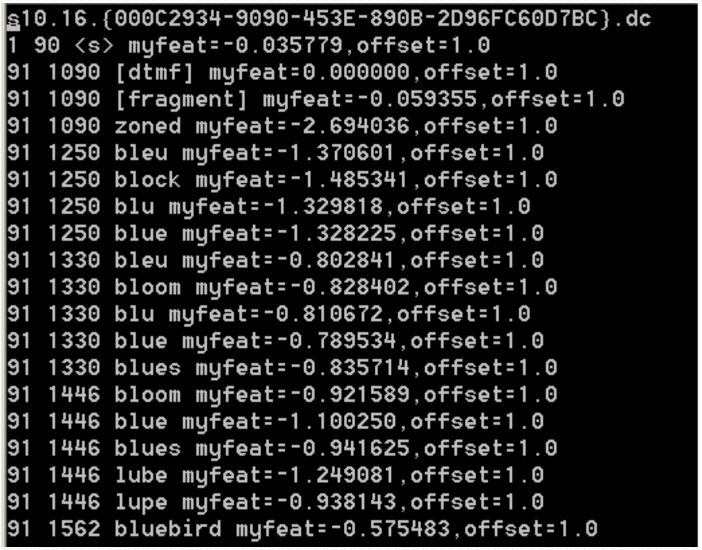
- Slide detectors for each multi-phone unit over speech
- Combine detections into  a single SCARF stream (unit-time pairs)



```
19960510_NPR_ATC#Ailene_Leblanc@0001.sd
# syl stream
!sent_start 43
DH@IY 7
T@UW 25
HH@EH 37
L@IH@K 54
AA@P@T 71
ER@Z 100
W@ER 115
P@AA@R@T 132
IH@S@IH@P 157
EY@T@IH@NG 180
IH@N 207
W@OW@R 219
G@EY@M@Z 245
!sent_end 1135
```

## Word Lattice Annotations

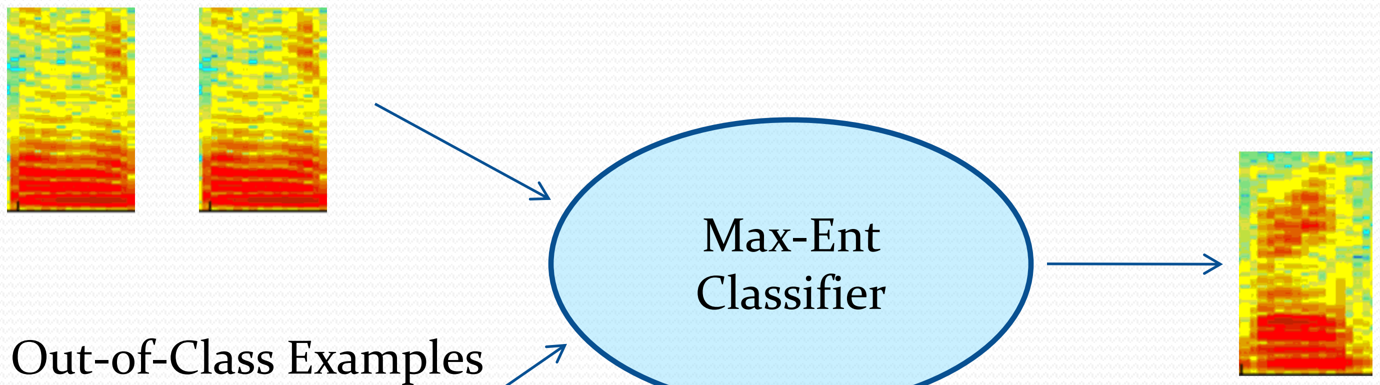
- Build window-based word models
- Provide alternative score as SCARF feature for each lattice link



```
s10.16.{000C2934-9090-453E-890B-2D96FC60D7BC}.dc
1 90 <s> myfeat=-0.035779,offset=1.0
91 1090 [dtmf] myfeat=0.000000,offset=1.0
91 1090 [fragment] myfeat=-0.059355,offset=1.0
91 1090 zoned myfeat=-2.694036,offset=1.0
91 1250 bleu myfeat=-1.370601,offset=1.0
91 1250 block myfeat=-1.485341,offset=1.0
91 1250 blu myfeat=-1.329818,offset=1.0
91 1250 blue myfeat=-1.328225,offset=1.0
91 1330 bleu myfeat=-0.802841,offset=1.0
91 1330 bloom myfeat=-0.828402,offset=1.0
91 1330 blu myfeat=-0.810672,offset=1.0
91 1330 blue myfeat=-0.789534,offset=1.0
91 1330 blues myfeat=-0.835714,offset=1.0
91 1446 bloom myfeat=-0.921589,offset=1.0
91 1446 blue myfeat=-1.100250,offset=1.0
91 1446 blues myfeat=-0.941625,offset=1.0
91 1446 lube myfeat=-1.249081,offset=1.0
91 1446 lupe myfeat=-0.938143,offset=1.0
91 1562 bluebird myfeat=-0.575483,offset=1.0
```

# STRF MPU Detectors

- Fixed window size for each multi-phone unit (median unit duration)
- Stacked acoustic feature vectors (VTLN+fMMI, Coherent/Convex modulation features [MF]) across window



In-Class Examples

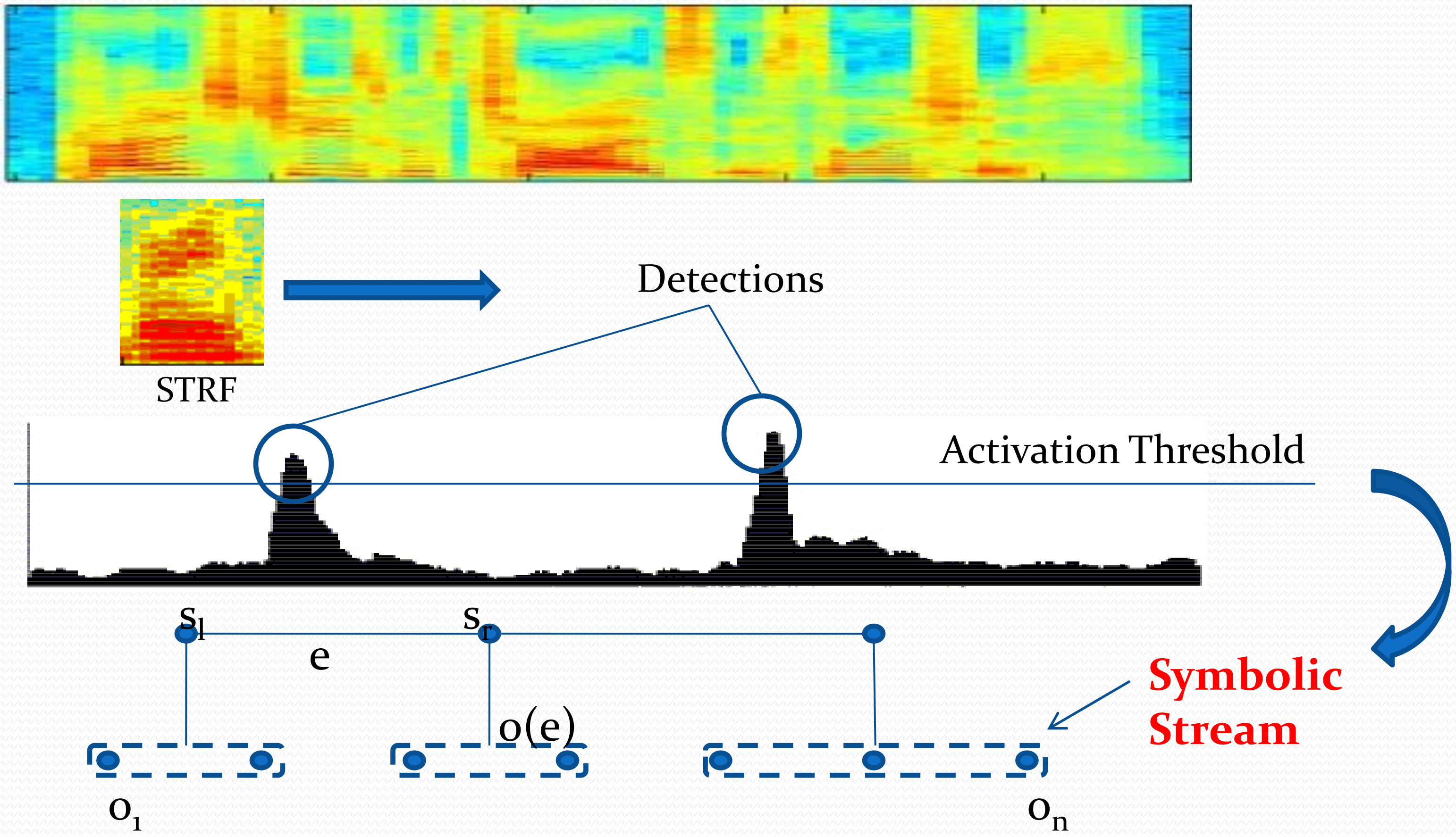
Out-of-Class Examples

Max-Ent Classifier

x 951: One for each unit

STRF for MPU:
ME Weight Matrix

# STRF MPU Detectors



STRF

Detections

Activation Threshold

$s_l$

$s_r$

$e$

$o(e)$

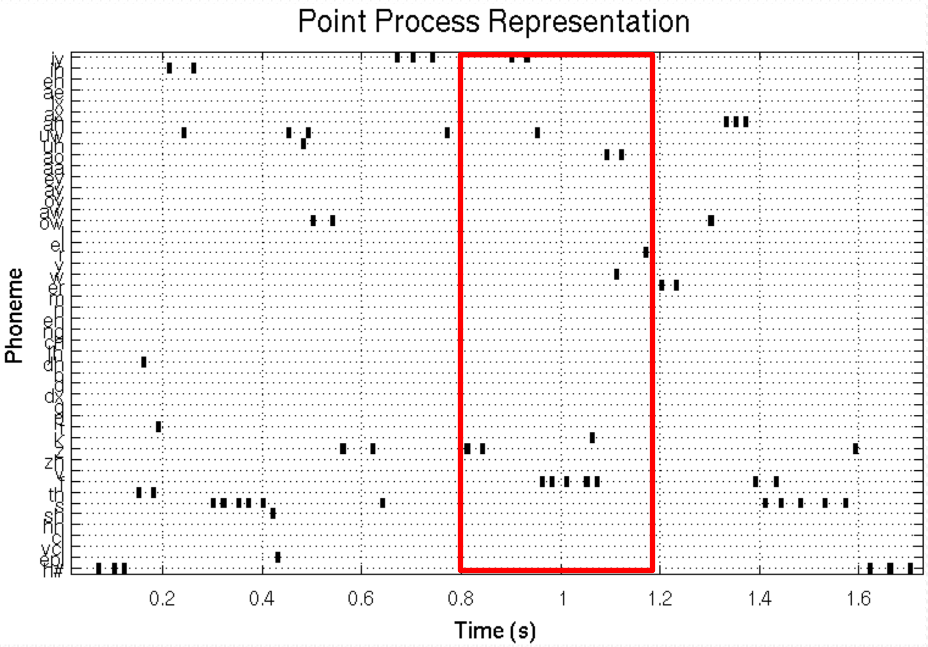
**Symbolic Stream**

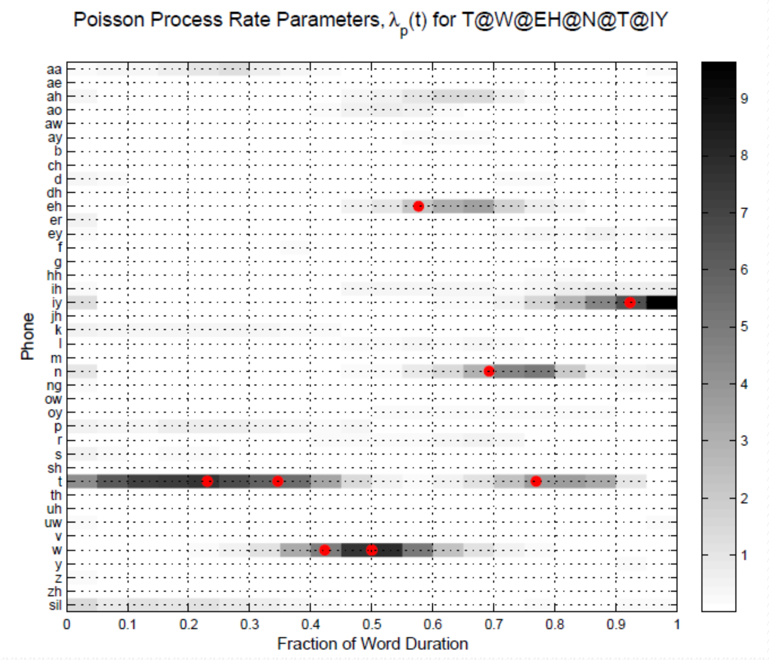
$o_1$

$o_n$

# PPM MPU Detectors

- Elastic windows normalized to unit duration (3982 units)
- Contained phone events modeled as inhomogeneous Poisson processes

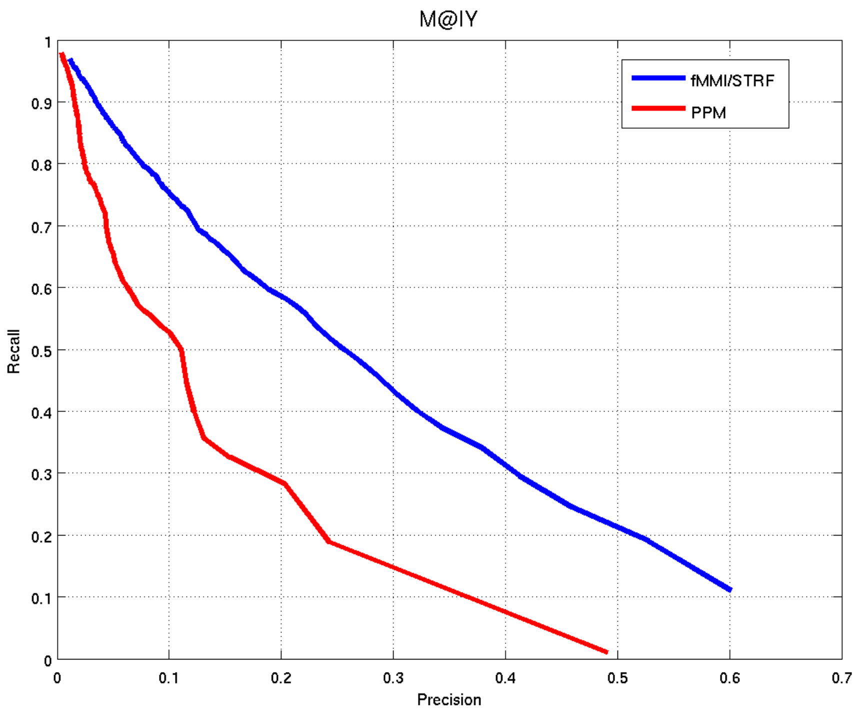




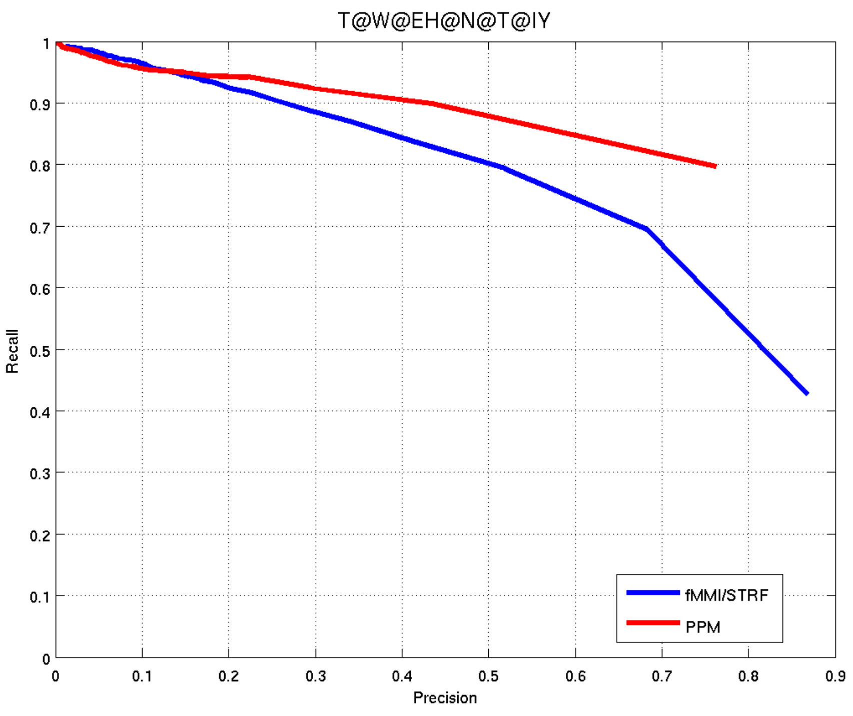
**An Aside:** Keith built a "zero resource" PPM-based keyword spotter that runs ~1000x faster than real time.

# MPU Detector Performance

## Short Unit: Me



## Longer Unit: Twenty



# MPU Detector Performance (cont'd)

| Features/Model | No. of Units | Avg. EER (%) |
|---|---|---|
| fMMI/STRF | 951 | 6.1 |
| Coherent MF/STRF | 951 | 20.8 |
| Convex MF/STRF | 951 | 18.2 |
| Phone Events/PPM | 3982 | 8.2 |

**Lessons learned:**

- VTLN+fMMI does adequate job of speaker normalization
- Fixed windows are adequate for shorter units
- Sparse representations are adequate for longer units
- Discriminative training is a good thing
- Our detectors did not improve upon SCARF baseline

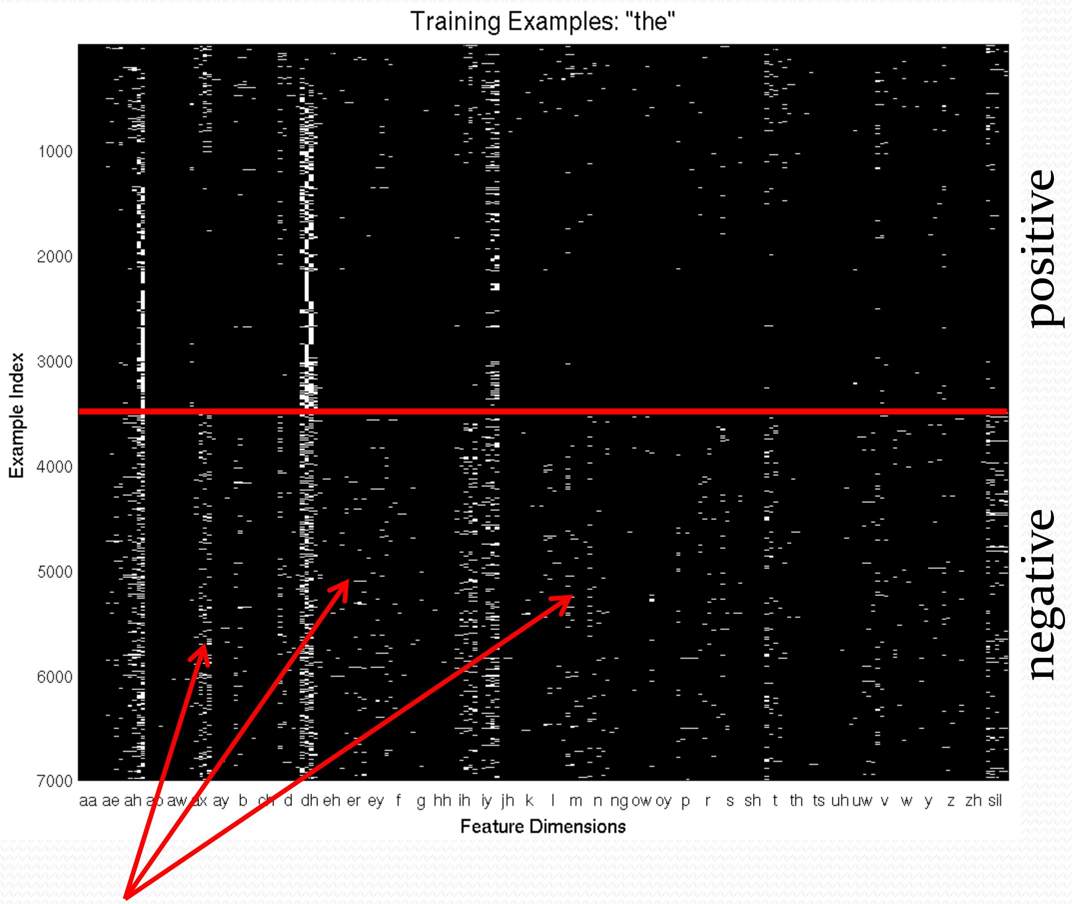
# STRF-Based Lattice Annotations

- 607 of the 1000 most frequent multi-phone units are words
- Use STRFs to classify the acoustics within each lattice arcs containing these **607 units**
- Use classifier scores as an additional SCARF feature for those arcs

**Note:** These one-vs-all classifiers are **trained across all units**

# PPM-Based Lattice Annotations

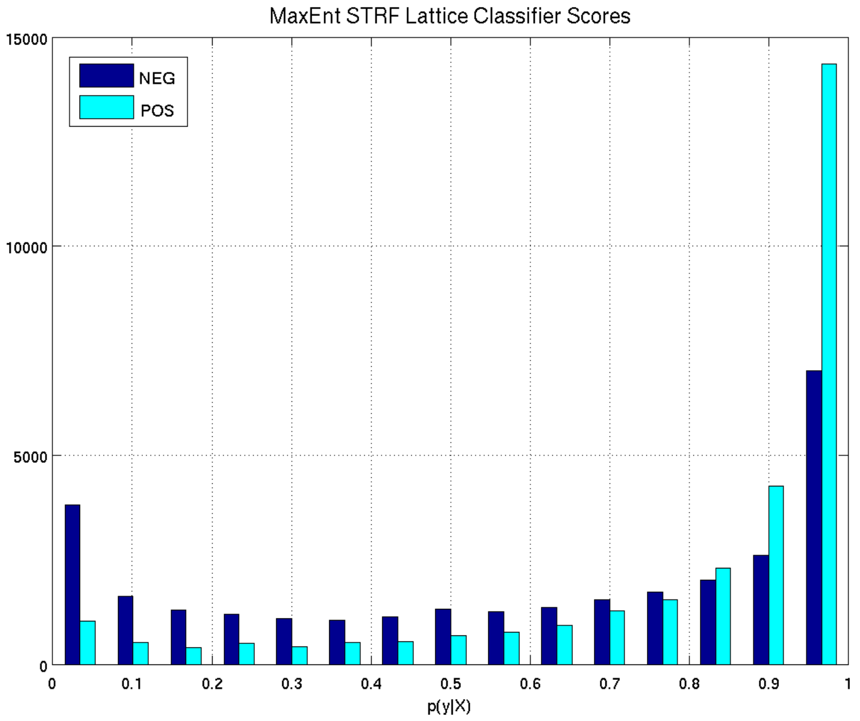
- Collect pos/neg point patterns for each word **from training lattices**
- Normalize all times to [0,1]
- Accumulate phone events in 10 bins → 420-dim space
- Rescore lattices with RLS+RBF classifiers for **top 72 error words**



Random phone events present in negative examples only

# Word Lattice Annotator Performance

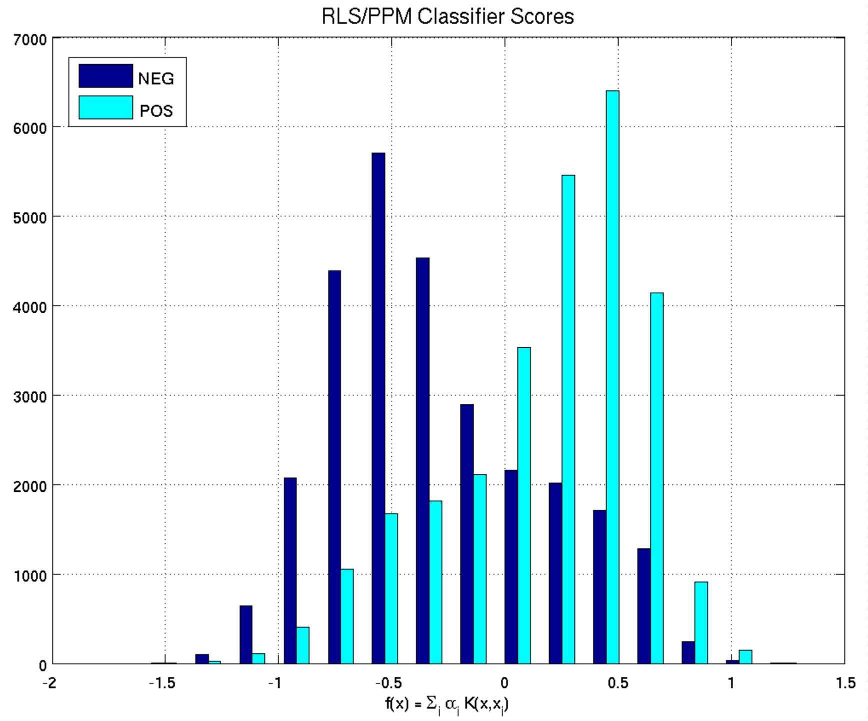
## fMMI/STRF Scores: “the”



**EER: 34.4%**

(trained on everything)

## PPM Scores: “the”



**EER: 26.0%**

(trained on lattice competitors only)

# SCARF Lattice Annotation Results

## Language Model Dependence (dev04f)

| | # Words | Unigram LM | Trigram LM |
|---|---|---|---|
| SCARF1 | --- | 16.9% WER | 16.0% WER |
| + fMMI/STRF Annotations | 607 | 16.3 | 15.9 |
| + PPM Annotations | 72 | 16.2 | 15.8 |

**Notice:** Lattice annotations provide from the acoustics most of what trigram LM does

# SCARF Lattice Annotation Results

## In Conjunction with MSR HMM Features (dev04f & rt04)

| Features | dev04f | rt04 (eval) |
|---|---|---|
| Baseline (Attila) | 16.3% WER | 15.7% WER |
| + SCARF retraining (SCARF-1) | 16.0 | 15.4 |
| + MSR HMM word annotations | 15.3 | 14.5 |
| + PPM 72 word annotations | **15.0** | **14.3** |
| (Lattice Oracle) | 11.2 | 10.1 |

SCARF+MSR+PPM →

8.0% relative gain
25% of possible gain
**dev04f**

8.9% relative gain
25% of possible gain
**rt04**

# Summary

- Investigated the role of window-based models in the SCARF framework
- Acoustic features + fixed window maximum entropy classifiers especially good for short, syllable-sized units
- Phone events + elastic window point process models especially good for longer multi-syllable units
- Discriminative training directly on the lattice competitors is a successful strategy for reducing errors
- Window-based lattice annotations led to improvements comparable to other workshop efforts

# Conclusion

Geoffrey Zweig

Patrick Nguyen

# Recap of Basic Idea

- SCARF enables us to unify the application of powerful new scientific approaches to ASR– e.g.
  - Template detections [Van Compernolle et al. 03]
  - Deep neural net features [Mohammed & Hinton 09]
  - Coherent modulation features [Atlas 09]
  - Point Process word models [Jansen 10]
  - Sparse Representation Phoneme Detectors [Hermansky et al. 10]
- At the workshop we pulled all this together and improved performance on two widely studied datasets

# Summary of Experiments

| **Wall Street Journal** | **Nov92** |
|---|---|
| Baseline (SPRAAK/HMM) | 7.3% WER |
| + SCARF, template features | 6.7 |
| (Lattice Oracle – best achievable) | 2.9 |

| **Broadcast News** | **Dev04f** |
|---|---|
| Baseline (Attila) | 16.3% WER |
| SCARF1 | 16.0 |
| +MSR word detectors | 15.3 |
| + TF-IDF, Duration, PPM, STRF, Phoneme detectors | **15.0** |
| (Lattice Oracle – best achievable) | 11.8 |

Significant gains on top of state-of-the-art systems

# Summary of Experiments

| Broadcast News | Dev04f | RT04f |
|---|---|---|
| Baseline (Attila) | 16.3% WER | 15.7% WER |
| SCARF1 | 16.0 | 15.4 |
| +MSR word detectors | 15.3 | 14.5 |
| +TF-IDF, Duration, PPM, STRF, Phoneme detectors | **15.0** | **14.2** |
| (Lattice Oracle) | 11.8 | 10.2 |

And results hold up on unseen test data – 9.6% relative improvement;

27% of possible gain achieved

# Summary of Accomplishments

- Created new framework of integrating diverse scientific advances in ASR
- Showed improvement on State-of-the-Art baselines for both Wall Street Journal and Broadcast News
- Fostered and integrated novel research on real-world tasks
  - Sparse Representation Phoneme Detectors
  - Deep Neural Nets
  - Point Process Models
  - Template features
  - Modulation representations

# Thank You

# References (1)

- SCARF
  - http://research.microsoft.com/en-us/projects/scarf/
  - G. Zweig and P. Nguyen, A Segmental CRF Approach to Large Vocabulary Continuous Speech Recognition, *ASRU* 2009
  - G. Zweig and P. Nguyen, SCARF: A Segmental Conditional Random Field Toolkit for Speech Recognition, INTERSPEECH 2010
  - G. Zweig, P. Nguyen, J. Droppo and A. Acero, Continuous Speech Recognition with a TF-IDF Acoustic Model, INTERSPEECH 2010
- MLPs
  - S. Thomas, S. Ganapathy and H. Hermansky, Phoneme Recognition Using Spectral Envelope and Modulation Frequency Features, ICASSP 2009
  - S. Thomas, S. Ganapathy and H. Hermansky, Tandem Representations of Spectral Envelope and Modulation Frequency Features for ASR, INTERSPEECH 2009

# References (2)

- Deep NNs
  - G. E. Hinton, S. Osindero, and Y. Teh, A fast learning algorithm for deep belief nets. Neural Computation, 18, pp 1527-1554, 2006
  - A. Mohamed, G. Dahl, G. E. Hinton,"Deep Belief Networks for phone recognition", in NIPS Workshop on Deep Learning for Speech Recognition and Related Applications, 2009
- Cohort Based Analysis
  - P. Xu, D. Karakos and S. Khudanpur, Self-Supervised Discriminative Training of Statistical Language Models, ASRU 2009
- Point Process Models
  - A. Jansen and P. Niyogi. Point Process Models for Spotting Keywords in Continuous Speech. IEEE Transactions on Audio, Speech, and Language Processing, 2009

# References (3)

- Modulation Features
  - P. Clark and L. Atlas, "Time-frequency coherent modulation filtering  of non-stationary signals," IEEE Trans. Signal Process., vol. 57, no. 11, pp. 4323-4332, 2009.
  - G. Sell and M. Slaney, "Solving Demodulation as an Optimization Problem," IEEE Trans. Signal Process., 2010
  - http://sites.google.com/a/uw.edu/isdl/projects/modulation-toolbox
  - http://ccrma.stanford.edu/~gsell/demodulation.html
- Template Recognition
  - M. De Wachter, M. Matton, K. Demuynck, P. Wambacq, R. Cools, and D. Van Compernolle. "Template-Based Continuous Speech Recognition." IEEE Transactions on Audio, Speech & Language Processing 15(4): 1377-1390, 2007
  - S. Demange and D. Van Compernolle. "HEAR: An Hybrid Episodic-Abstract speech Recognizer." INTERSPEECH 2009