
Word Recognition with Conditional Random Fields

Jeremy Morris

2/05/2010

language*speech*understa
og*generation*proc
translat
eech*understanding*lang
recognition*dialogue*re
eling*sys
understanding
eration*recogniti
ms*transl
language
ogue*generation*recogniti
translation*modeling*system
eech*understanding*lang
recognition*dialogue*genera

speech &
language
technologies
@osu

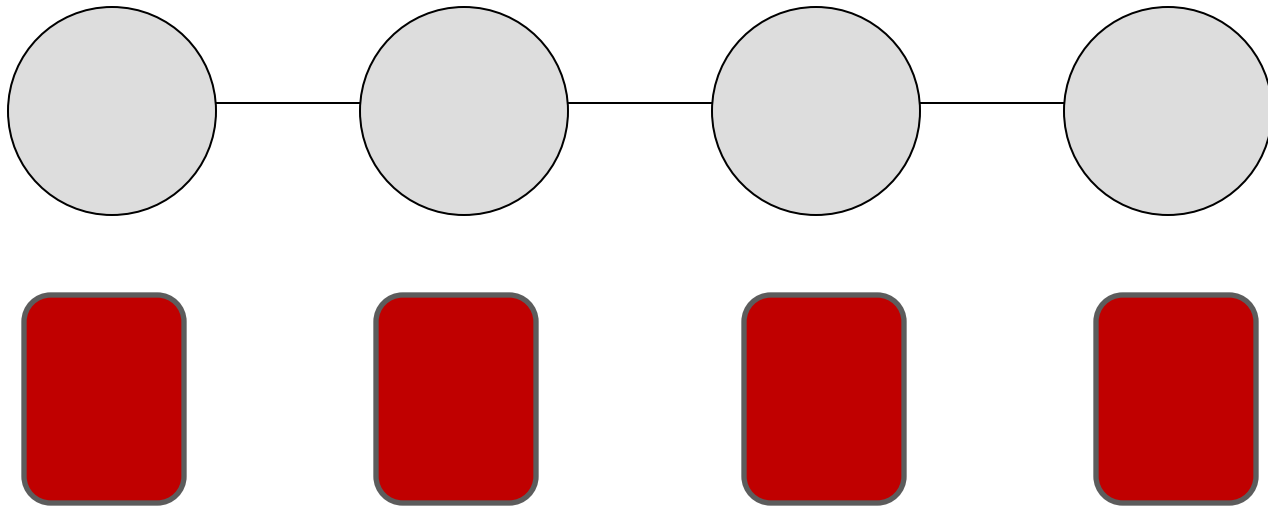


Outline

- Background
- Word Recognition – CRF Model
- Pilot System - TIDIGITS
- Larger Vocabulary - WSJ
- Future Work

Background

- Conditional Random Fields (CRFs)
 - Discriminative probabilistic sequence model
 - Directly defines a posterior probability $P(Y|X)$ of a label sequence Y given a set of observations X



Background

$$P(Y | X) = \frac{\exp \sum_k \left(\sum_i \lambda_i s_i(x, y_k) + \sum_j \mu_j t_j(x, y_k, y_{k-1}) \right)}{Z(x)}$$

- The form of the CRF model includes weighted *state feature functions* and weighted *transition feature functions*
 - Both types of functions can be defined to incorporate observed inputs

Background

- Our previous work compared CRF models for phone recognition to HMM models

Model	Accuracy
CRF (phone classes)	69.92%*
HMM Tandem16mix (phone classes)	69.34%
CRF (phone classes +phonological features)	70.63%*
HMM Tandem16mix (phone classes+ phonological features)	69.40%

*Significantly ($p < 0.05$) better than comparable Tandem 16mix triphone system (Morris & Fosler-Lussier 08)

Background

- Problem: How do we make use of CRF classification for word recognition?
 - Attempt to fit CRFs into current state-of-the-art models for speech recognition?
 - Attempt to use CRFs directly?
- Each approach has its benefits
 - Fitting CRFs into a standard framework lets us reuse existing code and ideas (Crandem system)
 - A model that uses CRFs directly opens up new directions for investigation
 - Requires some rethinking of the standard model for ASR

Review - Word Recognition

$$\arg \max_W P(W | X)$$

- Problem: For a given input signal X , find the word string W that maximizes $P(W|X)$

Review - Word Recognition

$$\arg \max_W P(W | X) = \arg \max_W \frac{P(X | W)P(W)}{P(X)}$$

- Problem: For a given input signal X , find the word string W that maximizes $P(W|X)$
- In an HMM, we would make this a generative problem

Review - Word Recognition

$$\arg \max_W P(W | X) = \arg \max_W P(X | W)P(W)$$

- Problem: For a given input signal X , find the word string W that maximizes $P(W|X)$
- In an HMM, we would make this a generative problem
- We can drop the $P(X)$ because it does not affect the choice of W

Review - Word Recognition

$$\arg \max_W P(W | X) = \arg \max_W P(X | W)P(W)$$

- We want to build phone models, not whole word models...

Review - Word Recognition

$$\begin{aligned}\arg \max_W P(W | X) &= \arg \max_W P(X | W)P(W) \\ &= \arg \max_W \sum_{\Phi} P(X | \Phi)P(\Phi | W)P(W)\end{aligned}$$

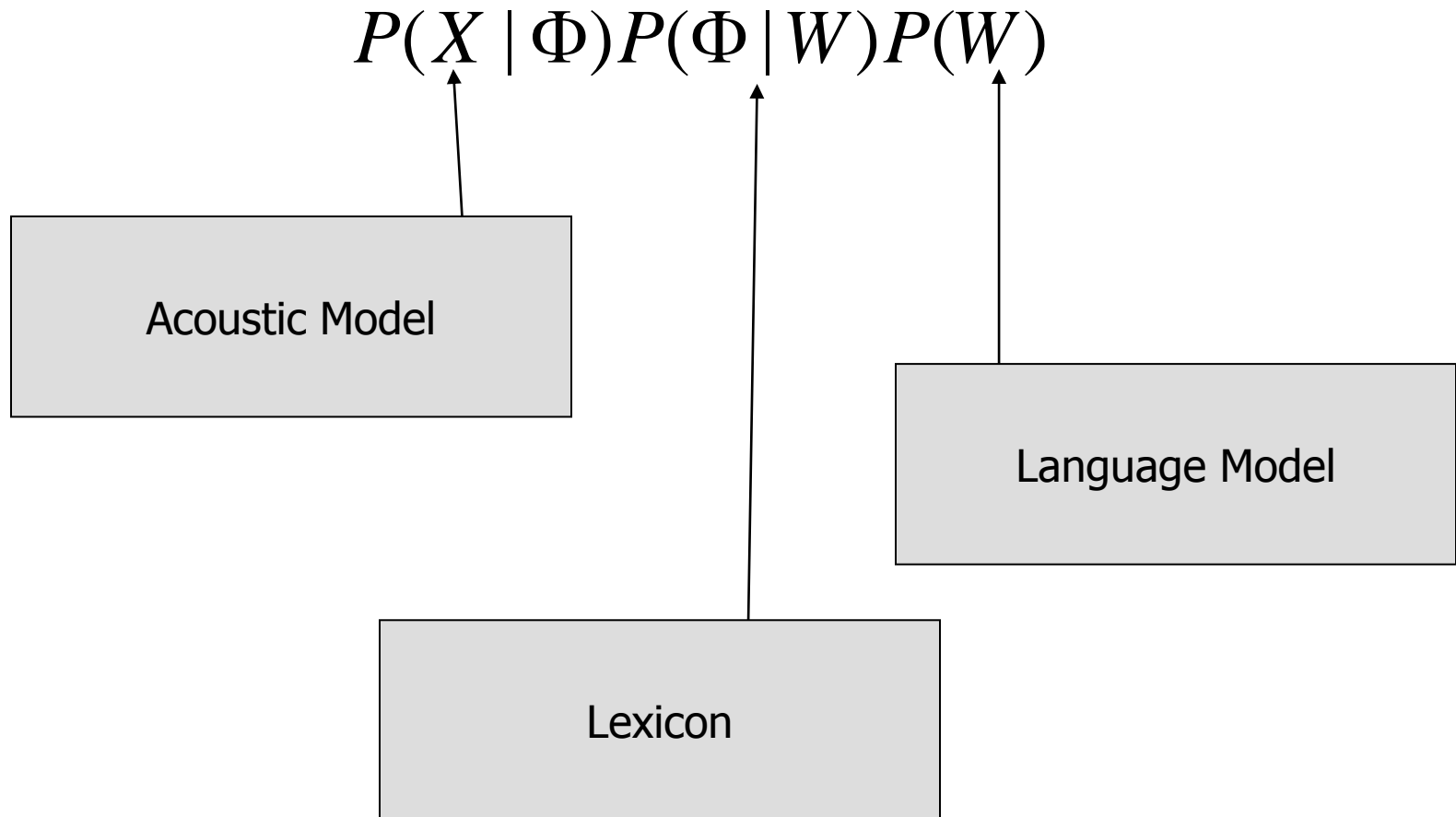
- We want to build phone models, not whole word models...
- ... so we marginalize over the phones

Review - Word Recognition

$$\begin{aligned}\arg \max_W P(W | X) &= \arg \max_W P(X | W)P(W) \\ &= \arg \max_W \sum_{\Phi} P(X | \Phi)P(\Phi | W)P(W) \\ &\approx \arg \max_{W, \Phi} P(X | \Phi)P(\Phi | W)P(W)\end{aligned}$$

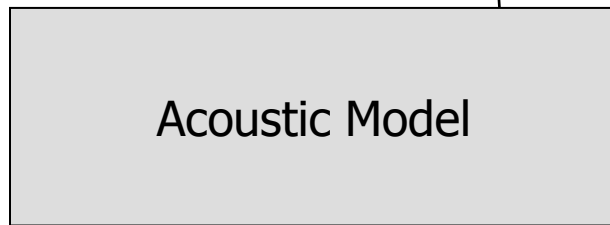
- We want to build phone models, not whole word models...
- ... so we marginalize over the phones
- and look for the best sequence that fits these constraints

Review - Word Recognition



Word Recognition

$$P(X | \Phi)P(\Phi | W)P(W)$$



- However - our CRFs model $P(\Phi|X)$ rather than $P(X|\Phi)$
 - This makes the formulation of the problem somewhat different

Word Recognition

$$\operatorname{arg\,max}_W P(W | X)$$

- We want a formulation that makes use of $P(\Phi|X)$

Word Recognition

$$\begin{aligned}\arg \max_W P(W | X) &= \arg \max_W \sum_{\Phi} P(W, \Phi | X) \\ &= \arg \max_W \sum_{\Phi} P(W | \Phi, X) P(\Phi | X)\end{aligned}$$

- We want a formulation that makes use of $P(\Phi|X)$
- We can get that by marginalizing over the phone strings
- But the CRF as we formulate it doesn't give $P(\Phi|X)$ directly

Word Recognition

$$P(W | \Phi, X)P(\Phi | X)$$

- Φ here is a *phone level* assignment of phone labels
- CRF gives related quantity – $P(Q|X)$ where Q is the *frame level* assignment of phone labels

Word Recognition

■ Frame level vs. Phone level

- Mapping from frame level to phone level may not be deterministic
- Example: The word “OH” with pronunciation /ow/
- Consider this sequence of frame labels:

OW OW OW OW OW OW OW

- This sequence can possibly be expanded many different ways for the word “OH” (“OH”, “OH OH”, etc.)

Word Recognition

- Frame level vs. Phone segment level
 - This problem occurs because we're using a single state to represent the phone /ow/
 - Phone either transitions to itself or transitions out to another phone
 - We can change our model to a multi-state model and make this decision deterministic
 - This brings us closer to a standard ASR HMM topology
ow1 ow2 ow2 ow2 ow2 ow3 ow3
 - Now we can see a single "OH" in this utterance

Word Recognition

$$\begin{aligned} P(\Phi | X) &= \sum_Q P(\Phi, Q | X) \\ &= \sum_Q P(\Phi | Q, X) P(Q | X) \\ &\approx \sum_Q P(\Phi | Q) P(Q | X) \end{aligned}$$

- Multi-state model gives us a deterministic mapping of $Q \rightarrow \Phi$
 - Each frame-level assignment Q has exactly one segment level assignment associated with it
 - Potential pitfalls if the multi-state model is inappropriate for the features we are using

Word Recognition

$$\begin{aligned}\arg \max_W P(W | X) &= \arg \max_W \sum_{\Phi} P(W | \Phi, X) P(\Phi | X) \\ &\approx \arg \max_W \sum_{\Phi, Q} P(W | \Phi, X) P(\Phi | Q) P(Q | X) \\ &\approx \arg \max_W \sum_{\Phi, Q} P(W | \Phi) P(\Phi | Q) P(Q | X)\end{aligned}$$

- Replacing $P(\Phi|X)$ we now have a model with our CRF in it
- What about $P(W| \Phi, X)$?
 - Conditional independence assumption gives $P(W| \Phi)$

Word Recognition

$$P(W | X) \approx \sum_{\Phi, Q} P(W | \Phi) P(\Phi | Q) P(Q | X)$$

- What about $P(W|\Phi)$?
 - Non-deterministic across sequences of words
 - $\Phi = / \text{ ah f eh r } /$
 - $W = ?$ “a fair”? “affair”?
 - The more words in the string, the more possible combinations can arise

Word Recognition

$$P(W | \Phi) = \frac{P(\Phi | W)P(W)}{P(\Phi)}$$

■ Bayes Rule

- $P(W)$ –language model
- $P(\Phi|W)$ – dictionary model
- $P(\Phi)$ – prior probability of phone sequences

Word Recognition

- What is $P(\Phi)$?
 - Prior probability over possible phone sequences
 - Essentially acts as a “phone fertility/penalty” term – lower probability sequences get a larger boost in weight than higher probability sequences
 - Approximate this with a standard n-gram model
 - Seed it with phone-level statistics drawn from the same corpus used for our language model

Word Recognition

$$\arg \max_W P(W | X) \approx \arg \max_{W, \Phi, Q} \frac{P(\Phi | W)P(W)}{P(\Phi)} P(\Phi | Q)P(Q | X)$$

- Our final model incorporates all of these pieces together
- Benefit of this approach – reuse of standard models
 - Each element can be built as a finite state machine (FSM)
 - Evaluation can be performed via FSM composition and best path evaluation as for HMM-based systems (Mohri & Riley, 2002)

Pilot Experiment: TIDIGITS

- First word recognition experiment – TIDIGITS recognition
 - Both isolated and strings of spoken digits, ZERO (or OH) to NINE
 - Male and female speakers
- Training set – 112 speakers total
 - Random selection of 11 speakers held out as development set
 - Remaining 101 speakers used for training as needed

Pilot Experiment: TIDIGITS

$$\arg \max_W P(W | X) \approx \arg \max_{W, \Phi, Q} \frac{P(\Phi | W)P(W)}{P(\Phi)} P(\Phi | Q)P(Q | X)$$

- Important characteristics of the DIGITS problem:
 - A given phone sequence maps to a single word sequence
 - A uniform distribution over the words is assumed
- $P(W|\Phi)$ easy to implement directly as FSM

Pilot Experiment: TIDIGITS

■ Implementation

- Created a composed dictionary and language model FST
 - No probabilistic weights applied to these FSTs – assumption of uniform probability of any digit sequence
- Modified CRF code to allow composition of above FST with phone lattice
 - Results scored using standard HTK tools
 - Compared to a baseline HMM system trained on the same features

Pilot Experiment: TIDIGITS

■ Labels

- Unlike TIMIT, TIDIGITS is only labeled at the word level
- Phone labels were generated by force aligning the word labels using an HMM-trained, MFCC based system

■ Features

- TIMIT-trained MLPs applied to TIDIGITS to create features for CRF and HMM training

Pilot Experiment: Results

Model	WER
HMM (triphone, 1 Gaussian, ~4500 parameters)	1.26%
HMM (triphone, 16 Gaussians ~120,000 parameters)	0.57%
CRF (monophone, ~4200 parameters)	1.11%
CRF (monophone, windowed, ~37000 parameters)	0.57%
HMM (triphone, 16 Gaussians, MFCCs)	0.25%

- Basic CRF performance falls in line with HMM performance for a single Gaussian model
- Adding more parameters to the CRF enables the CRF to perform as well as the HMM on the same features

Larger Vocabulary

- Wall Street Journal 5K word vocabulary task
 - Bigram language model
 - MLPs trained on 75 speakers, 6488 utterances
 - Cross-validated on 8 speakers, 650 utterances
 - Development set of 10 speakers, 368 utterances for tuning purposes
- Results compared to HMM-Tandem baseline and HMM-MFCC baseline

Larger Vocabulary

- Phone penalty model $P(\Phi)$
 - Constructed using the transcripts and the lexicon
 - Currently implemented as a phone pair (bigram) model
 - More complex model might lead to better estimates

Larger Vocabulary

- Direct finite-state composition not feasible for this task
 - State space grows too large too quickly
- Instead Viterbi decoding performed using the weighted finite-state models as constraints
 - Time-synchronous beam pruning used to keep time and space usage reasonable

Larger Vocabulary – Initial Results

Model	WER
HMM MFCC Baseline	9.3%
HMM PLP Baseline	9.7%
HMM Tandem MLP	9.1%
CRF (phone)	11.3%
CRF (phone windowed)	11.7%
CRF (phone + phonological)	10.9%
CRF (3state phone inputs)	12.4%
CRF (3state phone + phono)	11.7%
HMM PLP (monophone labels)	17.5%

- Preliminary numbers reported on development set only

Next Steps

- Context
 - Exploring ways to put more context into the CRF, either at the label level or at the feature level
- Feature selection
 - Examine what features will help this model, especially features that may be useful for the CRF that are not useful for HMMs
- Phone penalty model
 - Results reported with just a bigram phone model
 - A more interesting model leads to more complexity but may lead to better results
 - Currently examining trigram phone model to test the impact

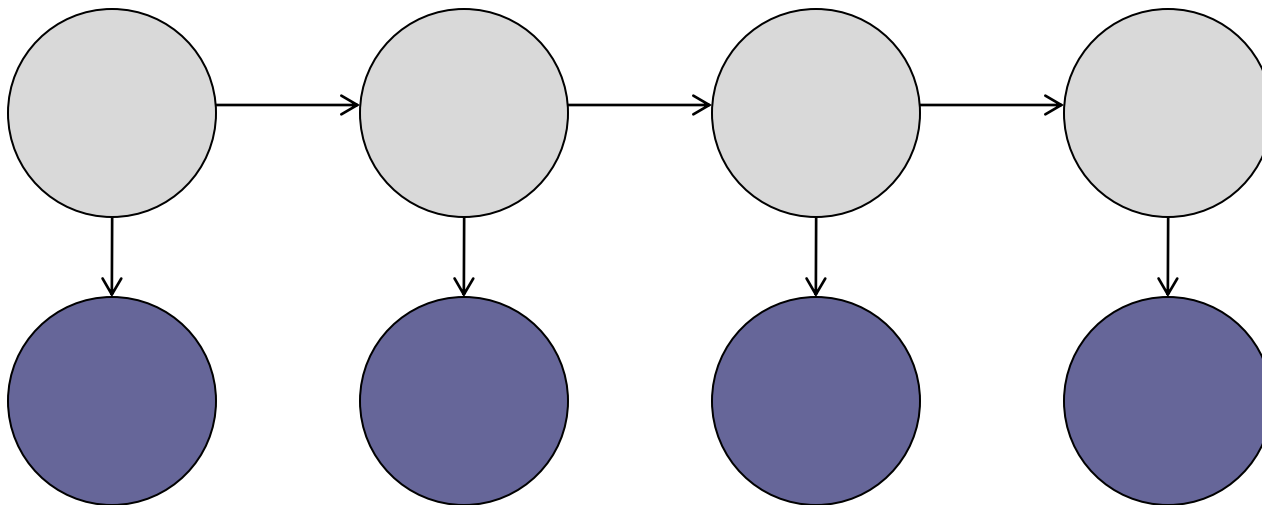
Discussion

References

- J. Lafferty et al, “Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data”, Proc. ICML, 2001
- A. Gunawardana et al, “Hidden Conditional Random Fields for phone classification”, Proc. Interspeech, 2005
- J. Morris and E. Fosler-Lussier. “Conditional Random Fields for Integrating Local Discriminative Classifiers”, IEEE Transactions on Audio, Speech and Language Processing, 2008
- M. Mohri et al, “Weighted finite-state transducers in speech recognition”, Computer Speech and Language, 2002

Background

- Tandem HMM
 - Generative probabilistic sequence model
 - Uses outputs of a discriminative model (e.g. ANN MLPs) as input feature vectors for a standard HMM



Background

- Tandem HMM
 - ANN MLP classifiers are trained on labeled speech data
 - Classifiers can be phone classifiers, phonological feature classifiers
 - Classifiers output posterior probabilities for each frame of data
 - E.g. $P(Q | X)$, where Q is the phone class label and X is the input speech feature vector

Background

■ Tandem HMM

- Posterior feature vectors are used by an HMM as inputs
- In practice, posteriors are not used directly
 - Log posterior outputs or “linear” outputs are more frequently used
 - “linear” here means outputs of the MLP with no application of a softmax function
 - Since HMMs model phones as Gaussian mixtures, the goal is to make these outputs look more “Gaussian”
 - Additionally, Principle Components Analysis (PCA) is applied to features to decorrelate features for diagonal covariance matrices

Idea: Crandem

- Use a CRF model to create inputs to a Tandem-style HMM
 - CRF labels provide a better per-frame accuracy than input MLPs
 - We've shown CRFs to provide better phone recognition than a Tandem system with the same inputs
- This suggests that we may get some gain from using CRF features in an HMM

Idea: Crandem

- Problem: CRF output doesn't match MLP output
 - MLP output is a per-frame vector of posteriors
 - CRF outputs a probability across the entire sequence
- Solution: Use Forward-Backward algorithm to generate a vector of posterior probabilities

Forward-Backward Algorithm

- Similar to HMM forward-backward algorithm
- Used during CRF training
- Forward pass collects feature functions for the timesteps prior to the current timestep
- Backward pass collects feature functions for the timesteps following the current timestep
- Information from both passes are combined together to determine the probability of being in a given state at a particular timestep

Forward-Backward Algorithm

$$P(y_{i,t} | X) = \frac{\alpha_{i,t} \beta_{i,t}}{Z(x)}$$

- This form allows us to use the CRF to compute a vector of local posteriors y at any timestep t .
- We use this to generate features for a Tandem-style system
 - Take log features, decorrelate with PCA

Phone Recognition

- Pilot task – phone recognition on TIMIT
 - 61 feature MLPs trained on TIMIT, mapped down to 39 features for evaluation
 - Crandem compared to Tandem and a standard PLP HMM baseline model
 - As with previous CRF work, we use the outputs of an ANN MLP as inputs to our CRF
- Phone class attributes
 - Detector outputs describe the phone label associated with a portion of the speech signal
 - /t/, /d/, /aa/, etc.

Results (Fosler-Lussier & Morris 08)

Model	Phone Accuracy
PLP HMM reference	68.1%
Tandem	70.8%
CRF	69.9%
Crandem – log	71.1%

* Significantly ($p < 0.05$) improvement at 0.6% difference between models

Word Recognition

- Second task – Word recognition on WSJ0
 - Dictionary for word recognition has 54 distinct phones instead of 48
 - New CRFs and MLPs trained to provide input features
 - MLPs and CRFs trained on WSJ0 corpus of read speech
 - No phone level assignments, only word transcripts
 - Initial alignments from HMM forced alignment of MFCC features
 - Compare Crandem baseline to Tandem and original MFCC baselines

Initial Results

Model	WER
MFCC HMM reference	9.12%
Tandem MLP (39)	8.95%
Crandem (19) (1 epoch)	8.85%
Crandem (19) (10 epochs)	9.57%
Crandem (19) (20 epochs)	9.98%

* Significant ($p \leq 0.05$) improvement at roughly 1% difference between models

Word Recognition

- CRF performs about the same as the baseline systems
- But further training of the CRF tends to degrade the result of the Crandem system
 - Why?
 - First thought – maybe the phone recognition results are deteriorating (overtraining)

Initial Results

Model	Phone Accuracy
MFCC HMM reference	70.09%
Tandem MLP (39)	75.58%
Crاندem (19) (1 epoch)	72.77%
Crاندem (19) (10 epochs)	72.81%
Crاندem (19) (20 epochs)	72.93%

* Significant ($p \leq 0.05$) improvement at roughly 0.07% difference between models

Word Recognition

- Further training of the CRF tends to degrade the result of the Crandem system
 - Why?
 - First thought – maybe the phone recognition results are deteriorating (overtraining)
 - Not the case
 - Next thought – examine the pattern of errors between iterations

Initial Results

Model	Total Errors	Insertions	Deletions	Subs.
Crandem (1 epoch)	542	57	144	341
Crandem (10 epochs)	622	77	145	400
Shared Errors	429	37	131* (102)	261** (211)
New Errors (1->10)	193	40	35	118

* 29 deletions are substitutions in one model and deletions in the other

**50 of these subs are different words between the epoch 1 and epoch 10 models

Word Recognition

- Training the CRF tends to degrade the result of the Crandem system
 - Why?
 - First thought – maybe the phone recognition results are deteriorating (overtraining)
 - Not the case
 - Next thought – examine the pattern of errors between iterations
 - There doesn't seem to be much of a pattern here, other than a jump in substitutions
 - Word identity doesn't give a clue – similar words wrong in both lists

Word Recognition

- Further training of the CRF tends to degrade the result of the Crandem system
 - Why?
 - Current thought – perhaps the reduction in scores of the correct result is impacting the overall score
 - This appears to be happening in at least some cases, though it is not sufficient to explain everything

Word Recognition

MARCH vs. LARGE

Iteration 1

0	0	m	0.952271	I	0.00878177	en	0.00822043	em	0.00821897
0	1	m	0.978378	em	0.00631441	I	0.00500046	en	0.00180805
0	2	m	0.983655	em	0.00579973	I	0.00334182	hh	0.00128429
0	3	m	0.980379	em	0.00679143	I	0.00396782	w	0.00183199
0	4	m	0.935156	aa	0.0268882	em	0.00860147	I	0.00713632
0	5	m	0.710183	aa	0.224002	em	0.0111564	w	0.0104974 I 0.009005

Iteration 10

0	0	m	0.982478	em	0.00661739	en	0.00355534	n	0.00242626 I 0.001504
0	1	m	0.989681	em	0.00626308	I	0.00116445	en	0.0010961
0	2	m	0.991131	em	0.00610071	I	0.00111827	en	0.000643053
0	3	m	0.989432	em	0.00598472	I	0.00145113	aa	0.00127722
0	4	m	0.958312	aa	0.0292846	em	0.00523174	I	0.00233473
0	5	m	0.757673	aa	0.225989	em	0.0034254	I	0.00291158

Word Recognition

MARCH vs. LARGE - logspace

Iteration 1

0	0	m	-0.0489053	l	-4.73508	en	-4.80113	em	-4.80131		
0	1	m	-0.0218596	em	-5.06492	l	-5.29822	en	-6.31551		
0	2	m	-0.01648	em	-5.14994	l	-5.70124	hh	-6.65755		
0	3	m	-0.0198163	em	-4.99209	l	-5.52954	w	-6.30235		
0	4	m	-0.0670421	aa	-3.61607	em	-4.75582	l	-4.94256		
0	5	m	-0.342232	aa	-1.4961	em	-4.49574	w	-4.55662	l	-4.71001

Iteration 10

0	0	m	-0.017677	em	-5.01805	en	-5.6393	n	-6.02141	l	-6.49953
0	1	m	-0.0103729	em	-5.07308	l	-6.75551	en	-6.816		
0	2	m	-0.0089087	em	-5.09935	l	-6.79597	en	-7.34928		
0	3	m	-0.0106245	em	-5.11855	l	-6.53542	aa	-6.66307		
0	4	m	-0.0425817	aa	-3.53069	em	-5.25301	l	-6.05986		
0	5	m	-0.277504	aa	-1.48727	em	-5.67654	l	-5.83906		

Word Recognition

■ Additional issues

- Crandem results sensitive to format of input data
 - Posterior probability inputs to the CRF give very poor results on word recognition.
 - I suspect is related to the same issues described previously
- Crandem results also require a much smaller vector after PCA
 - MLP uses 39 features – Crandem only does well once we reduce to 19 features
 - However, phone recognition results improve if we use 39 features in the Crandem system (72.77% -> 74.22%)