Interspeech 2013 Tutorial:
Advances in Large Vocabulary Continuous Speech Recognition

George Saon† and Jen-Tzung Chien‡

†IBM T. J. Watson Research Center, Yorktown Heights, NY
‡National Chiao Tung University, Hsinchu, Taiwan
Introduction

• Several advances have been made to the design of modern LVCSR systems over the past decade

• Successful applications: smartphones (Siri, Google now), open-domain voice search, in-car speech recognition (e.g. navigation), broadcast news transcription, meeting transcription, conversational telephone speech transcription, call center applications, speech-to-speech translation, medical & legal transcription, dictation, etc.

• Problem is far from being solved: background noise, channel distortions, foreign accents, disfluent speech, topic change can cause recognition errors
Scope of this tutorial

- Focus on successful techniques used in U.S. government-led speech recognition evaluations (HUB4, HUB5, EARS, GALE, RATS, BABEL) (Saon & Chien 2012c)

- Incorporated in LVCSR systems from universities such as Cambridge (UK), LIMSI (France), RWTH Aachen (Germany), CMU (USA) and commercial institutions like AT&T, BBN, Google, IBM, Microsoft, and SRI

- Limit the discussion to language-independent algorithms
LVCSR system

\[ \hat{W} = \arg\max_{W \in \mathcal{V}^*} p(W|X) = \arg\max_{W \in \mathcal{V}^*} p(X|W) p(W) \]

size of \( \mathcal{V} \) on the order of \( 10^4 \ldots 10^6 \) words
Layered architecture

(figure courtesy of S. Young)
Table of contents

• Introduction

• **Front-end processing**

• Acoustic modeling

• Deep neural networks

• Language modeling

• Hypothesis search and system combination

• New directions and conclusions
Front-end processing

- Feature Extraction and Transformation
  - MFCC, PLP
  - Mean and Var Normalization
  - Delta, Delta-Delta Features
  - LDA, HDA, HLDA
- Noise Robust Features
  - SPLICE
  - QE
- Adaptive and Discriminative Features
  - VTLN, fMLLR
  - fMPE, NN

Speech Signal → Feature Vectors
Feature extraction

- Short-term fast Fourier transform (FFT) of the speech signal within a 25ms window every 10ms

- Energies of the neighboring frequencies within each frame are binned together via a mel-scale filterbank

- Log mel-spectra are decorrelated via a discrete cosine transform (DCT) yielding a 13-dimensional vector of mel frequency cepstral coefficients (MFCCs)

- MFCCs have been replaced with a more noise-robust representation based on perceptual linear prediction (PLP) coefficients (Hermansky 1990)
MFCC pipeline
Normalization and transformation

- Utterance-based CMS and speaker-based CVN

- delta and delta-delta coefficients (Furui 1986) have been replaced by a linear projection matrix such as linear discriminant analysis (LDA) (Fukunaga 1990)

- LDA features are further rotated by a semi-tied covariance (STC) transform (Gales 1998)

- LDA+STC can reduce WER by 10-15% over $\Delta$, $\Delta\Delta$
Modeling temporal changes

- Construct a supervector that concatenates $N$ (say 9) consecutive frames

- Project the resulting supervector to a lower dimensionality

- Linear transforms can be used
  - to simulate least squares estimate for temporal derivatives or
  - to maximize the discrimination between phonetic classes
Linear discriminant analysis

Class definition important for LDA: HMM state-level classes are best
Heteroscedastic discriminant analysis

(Kumar & Andreou 1998, Saon et al. 2000)
**Semi-tied covariance transform**

Problem: final features modeled with **diagonal covariance** Gaussians

STC transform rotates the feature space so that the diagonal covariance assumption is more valid (Gales 1999)
Speaker-adaptive features

- Intra-speaker variations & inter-speaker variations

- Feature-domain speaker normalization techniques produce a canonical feature space by eliminating inter-speaker variability
  - warping the frequency axis to match the vocal tract length of a reference speaker as in VTLN (Wegmann et al. 1996, Lee & Rose 1998)
  - affinely transforming the features to maximize the likelihood under the current model as in feature-space maximum likelihood linear regression (fMLLR) (Gales 1998)
  - feature-space Gaussianization (Saon et al. 2004) of empirical distribution of adaptation data
  - VTLN+fMLLR+MLLR can reduce WER by 20%-30%
VTLN

- Make speech from all speakers appear as if it was produced by a vocal tract of a single standard length

- Optimal frequency warping factor $\hat{\alpha}_s$ for speaker $s$ is obtained by maximizing the likelihood of the warped utterances $X_s(\alpha)$ in frequency domain with respect to the acoustic model $\lambda$ and the transcriptions $W_s$

  $$\hat{\alpha}_s = \arg \max_{\alpha} p(X_s(\alpha)|\lambda, W_s)$$

  where $\lambda$ is trained from a large population of speakers

- Warped utterance $X_s(\hat{\alpha}_s)$ is decoded to find the recognition result
VTLN transfer function

Piecewise linear transfer function allowing for a ±20% warping
VTLN - warped filterbanks
VTLN - Jacobian compensation

- Sequence of original cepstra $\mathbf{X} = \mathbf{x}_1, \ldots, \mathbf{x}_T$

- Sequence of warped cepstra $\mathbf{X}(\alpha) = \mathbf{x}_1(\alpha), \ldots, \mathbf{x}_T(\alpha), \alpha \in [0.8, 1.2]$

- $\mathbf{x}_t$ and $\mathbf{x}_t(\alpha)$ related through a non-linear mapping $T_\alpha$

  $$\mathbf{x}_t(\alpha) = T_\alpha(\mathbf{x}_t)$$

- Jacobian of the feature transformation $\mathbf{x}_t \mapsto \mathbf{x}_t(\alpha)$

  $$\left( \frac{\partial T_\alpha(\mathbf{x})_i}{\partial \mathbf{x}_j} \right)$$
• **Assumption:** \( T_\alpha = A_\alpha \) with \( a \in \mathbb{R}^{n \times n} \)

• It follows that
\[
\left| \frac{\partial T_\alpha(x)}{\partial x} \right| = | A_\alpha |
\]

• No need to know \( A_\alpha \) because
\[
\log \left| \sum_{t=1}^{T} x_t(\alpha) x_t(\alpha)^T \right| = 2 \log | A_\alpha | + \log \left| \sum_{t=1}^{T} x_t x_t^T \right|
\]

• Jacobian compensation scheme is to add
\[
\frac{1}{2} \log \left| \sum_{t=1}^{T} x_t(\alpha) x_t(\alpha)^T \right|
\]

to the average log-likelihood of the warped cepstra (Saon *et al.* 2003)
VTLN - warp factor distribution example
fMLLR adaptation

- Affinely transform the features $X = \{x_t\}_{t=1}^T$ to maximize the likelihood under the current model

- **Affine** transformation of a feature vector $x_t \in \mathbb{R}^D$ using speaker-specific matrix parameter $M^f = [A^T \ b^T]^T$ is performed by

  $\hat{x}_t = Ax_t + b = M^f \xi_t$

where $\xi_t = [x_t^T \ 1]^T$ denotes the extended feature vector

- fMLLR is equivalent to **constrained** (model-space) MLLR
Gaussianization

- **Gaussianization** (Saon et al. 2004) aims to compensate the mismatch between reference Gaussian distribution and empirical distribution of adaptation data.

- **Dimension-wise nonlinear** feature transformation $T : \mathbb{R}^D \rightarrow \mathbb{R}^D$

  $$y = T(x) = (G^{-1} \circ F_N)(x)$$

- $F_N(x_i) = \frac{\text{rank}(x_i)}{N}$, empirical CDF of adaptation data $x_1, \ldots, x_N$

- $G(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$, Gaussian CDF
Gaussianization example

original distribution

transformed distribution
Gaussianizing transform
Discriminative features

- **Feature-space minimum phone error** (fMPE) (Povey *et al.* 2005) is a transformation that provides time-dependent offsets to the regular feature vectors.

- **fMPE+MPE** can reduce WER by 25%.

- **Tandem processing** uses a multi-layer perceptron (MLP) to estimate phone posteriors which are modeled with GMMs (Hermansky *et al.* 2000).

- **Bottleneck features** extracted with a 5-layer neural network (NN) with a constriction layer (Grezl *et al.* 2007).
Bottleneck features

(Grezl & Fousek 2008)
Modern LVCSR front-end
Table of contents

- Introduction
- Front-end processing
- Acoustic modeling
- Deep neural networks
- Language modeling
- Hypothesis search and system combination
- New directions and conclusions
HMM acoustic modeling

- \( D \)-dimensional speech feature vectors \( X = \{x_t\}_{t=1}^T \) collected for estimation of HMM parameters \( \Lambda = \{\pi_i, a_{ij}, \omega_{ik}, \mu_{ik}, \Sigma_{ik}\} \)
  - \( \omega_{ik} \): mixture weight
  - \( \mu_{ik} \): mean vector
  - \( \Sigma_{ik} \): covariance matrix

\[
p(x_t|\Lambda_i) = \sum_{k=1}^{K} \omega_{ik} \mathcal{N}(x_t; \mu_{ik}, \Sigma_{ik})
\]

- Maximum likelihood (ML) model parameters are estimated by maximizing

\[
p(X|\Lambda) = \sum_{S=\{s_t\}} \left[ \pi_{s_1} p(x_1|\Lambda_{s_1}) \prod_{t=2}^{T} a_{s_{t-1}s_t} p(x_t|\Lambda_{s_t}) \right]
\]
ML estimation

• ML estimation

\[ \hat{\Lambda} = \arg \max_{\Lambda} p(X|\Lambda) \]

• Expectation-maximization (EM) algorithm (Dempster et al. 1977) is applied to solve ML estimation with missing variable \( S \)

\[ \Lambda^{(k+1)} = \arg \max_{\Lambda} Q(\Lambda|\Lambda^{(k)}) = \arg \max_{\Lambda} \sum_{S} p(S|X, \Lambda^{(k)}) \log p(X, S|\Lambda) \]
Acoustic modeling

Discriminative Training
- Training Criterion: MCE, MMI, MPE, MPFE, BMMI, PLM
- Feature Space and Model Space: RDLT, fMPE+MPE, fBMMI+BMMI

Speaker Adaptation
- Adaptation Criterion: ML, AAP, MMI, MPE, CML
- Feature Space and Model Space: fMLLR, MLLR, Eigenvoices, Eigen-MLLR
- Noise Adaptation: PMC, VTS, NAT, JUD

Deep Neural Networks DBN

Sequence of Feature Vectors → Acoustic Modeling → Acoustic Parameters
**MMI training**

- We aim to achieve the lowest WERs on unseen test data.

- **Maximum mutual information (MMI) criterion** (Bahl *et al.* 1986)

\[
F_{\text{MMI}}(\Lambda) \triangleq \mathcal{I}_\Lambda(X, W^r) = \log \frac{p_\Lambda(X, W^r)}{p_\Lambda(X)p(W^r)} \propto \log p_\Lambda(W^r|X)
\]

\[
= \log p_\Lambda(X|W^r) - \log \sum_W p_\Lambda(X|W)p(W)
\]

\[
\triangleq F^{\text{num}}(\Lambda) - F^{\text{den}}(\Lambda)
\]

- MMI training is interpreted as a maximization of log posterior probability \(\log p_\Lambda(W^r|X)\) of the correct word sequence \(W^r\).

- Same as Conditional maximum likelihood (CML).
MMI auxiliary function

- Extended Baum-Welch algorithm by maximizing the “weak-sense” auxiliary function

\[
Q(\Lambda | \Lambda^{(k)}) = \sum_{S} p(S | X, W^r, \Lambda^{(k)}) \log p(X, S | \Lambda) \underbrace{- \sum_{S} \sum_{W} p(S, W | X, \Lambda^{(k)}) \log p(X, S | \Lambda)}_{Q^{\text{num}}(\Lambda | \Lambda^{(k)})} + Q^{\text{sm}}(\Lambda | \Lambda^{(k)})
\]

- **Smoothing** function \(Q^{\text{sm}}(\Lambda | \Lambda^{(k)})\) is added to ensure that auxiliary function increases after each iteration
MMI solution

• MMI update for Gaussian means and covariance matrices (Povey & Woodland 2002)

\[
\hat{\mu}_{ik} = \frac{\{\theta_{ik}^{\text{num}}(x) - \theta_{ik}^{\text{den}}(x)\} + (D_{ik} + \tau)\mu_{ik}}{\{\gamma_{ik}^{\text{num}} - \gamma_{ik}^{\text{den}}\} + D_{ik} + \tau}
\]

\[
\hat{\Sigma}_{ik} = \frac{\{\theta_{ik}^{\text{num}}(xx^T) - \theta_{ik}^{\text{den}}(xx^T)\} + (D_{ik} + \tau)(\mu_{ik}\mu_{ik}^T + \Sigma_{ik})}{\{\gamma_{ik}^{\text{num}} - \gamma_{ik}^{\text{den}}\} + D_{ik} + \tau} - \hat{\mu}_{ik}\hat{\mu}_{ik}^T
\]

• I-smoothing factor \(\tau\) and Gaussian-specific empirical parameters \(D_{ik}\)

• \(\{\gamma_{ik}, \theta_{ik}(x), \theta_{ik}(xx^T)\}\) are the zero-order, first-order and second-order sufficient statistics
MPE training

- MCE estimation was designed to minimize the weighted sentence error rate or weighted word error rate (WER).

- Discriminative training based on the criterion of minimum phone error (MPE) (Povey & Woodland 2002) aims to minimize the expected phone error rate or maximize the expected phone accuracy.

\[ F_{\text{MPE}}(\Lambda) \triangleq \sum_{r=1}^{R} \sum_{W} p_{\Lambda}^k(W|X_r)A(W, W^r) \]

where \( A(W, W^r) \) is the raw phone accuracy of sentence \( W \) given the reference transcription \( W^r \) for the \( r \)th utterance.

- Minimum phone frame error (MPFE) (Zheng & Stolcke 2005) uses a frame-based phone accuracy measure which is easy to compute.
Feature-space MPE

- fMPE (Povey et al. 2005) or fMMI training is performed by transforming acoustic features to $\hat{x}_t = \{\hat{x}_{td}\}$ for each frame by $\hat{x}_t = x_t + M^f h_t$

  - $M^f = \{m^f_{dj}\}$: transformation matrix
  - $h_t = \{h_{tj}\}$: high dimensional Gaussian posteriors

$$m^f_{dj} \leftarrow m^f_{dj} + \eta_{dj} \frac{\partial Q}{\partial m^f_{dj}} = m^f_{dj} + \eta_{dj} \sum_t \frac{\partial Q}{\partial \hat{x}_{td}} h_{tj}$$

$$\frac{\partial}{\partial \hat{x}_t} Q(\hat{x}_t, \Lambda(\hat{x}_t)|\Lambda^{(k)}) = \frac{\partial Q}{\partial \hat{x}_t} \left( \begin{array}{c} \text{direct} \\ \text{indirect} \end{array} \right) + \frac{\partial Q}{\partial \Lambda} \frac{\partial \Lambda}{\partial \hat{x}_t}$$

- fMPE outperforms MPE for LVCSR
Region-dependent linear transform

- Feature transformation in fMPE is arranged as

\[
\hat{x}_t = x_t + M^f h_t = \sum_j h_{tj}(x_t + m_j)
\]

where \(m_j\) is the \(j\)th column of \(M^f\)

- RDLT (Zhang et al. 2006) is a generalization of fMPE by replacing it with the affine transformations

\[
\hat{x}_t = \sum_j h_{tj}(A_j x_t + b_j)
\]
BMMI training

- Boosted MMI (BMMI) (Povey et al. 2008) training considers the criterion with a boosting factor $b$

\[
\mathcal{F}_{\text{BMMI}}(\Lambda) \triangleq \sum_{r=1}^{R} \log \frac{p^\kappa_{\Lambda}(X_r|W^r)p(W^r)}{\sum_{W} p^\kappa_{\Lambda}(X_r|W)p(W) \exp(-bA(W, W^r))}
\]

- Boost the likelihood of the sentences that have more errors

- BMMI is linked to penalized large margin (PLM) (Sha & Saul 2007, Saon & Povey 2008)
Penalized large-margin

- PLM training (Saon & Povey 2008) solves a constrained margin maximization problem

\[
\max b \quad \text{s.t.} \quad \log p_\Lambda(X_r, W^r) - \log p_\Lambda(X_r, W) \geq bH(W, W^r), \ \forall W, 1 \leq r \leq R
\]

where \( H(W, W^r) \) is the frame-based Hamming distance between \( W \) and \( W^r \)

- Objective:

\[
\mathcal{F}_{PLM}(\Lambda, b) \triangleq b + \frac{1}{\rho} \sum_{r=1}^{R} \log \frac{p_\Lambda(X_r|W^r)p(W^r)}{\sum_W p_\Lambda(X_r|W)p(W) \exp(bH(W, W^r))}
\]

where \( \rho \) is a penalty or tradeoff parameter between margin and constraint
Feature-space and model-space training

- The same discriminative training criterion $Q(\{\hat{x}_t\}, \Lambda(\{\hat{x}_t\})|\Lambda^{(k)})$ is jointly optimized with respect to acoustic features $\{\hat{x}_t\}$ or $M^f$ and HMM parameters $\Lambda$

- $f$MPE+$MPE$ and $f$BMMI+$BMMI$ are compared in (Povey et al. 2008)

$f$BMMI+$BMMI$ has been shown to be superior to $f$MPE+$MPE$ for several LVCSR tasks.
Examples of WERs after discriminative training

<table>
<thead>
<tr>
<th>Training</th>
<th>English CTS 300h</th>
<th>English BN 50h</th>
<th>Arabic BN 1400h</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML</td>
<td>18.9%</td>
<td>22.5%</td>
<td>17.1%</td>
</tr>
<tr>
<td>fBMMI</td>
<td>16.4%</td>
<td>18.4%</td>
<td>14.3%</td>
</tr>
<tr>
<td>fBMMI+BMMI</td>
<td>15.1%</td>
<td>17.2%</td>
<td>12.6%</td>
</tr>
</tbody>
</table>
MLLR speaker adaptation

- **Maximum likelihood linear regression** (Leggetter & Woodland 1995) aims to transform the clusters of HMM mean vectors to a target speaker by using regression matrices

\[
\hat{\mu}_{ik} = A_c \mu_{ik} + b_c = M_c \xi_{ik}
\]

where \( M_c = [A^T_c \ b^T_c]^T \) and \( \xi_{ik} = [\mu^T_{ik} \ 1]^T \)

- Solution to MLLR adaptation is solved by EM algorithm

\[
\hat{M}^{(k+1)} = \{ \hat{M}^{(k+1)}_c \} = \arg \max_M Q(M|M^{(k)})
\]

\[
= \arg \max_M \sum_S p_\Lambda(S|X, M^{(k)}) \log p_\Lambda(X, S|M)
\]
fMLLR+MLLR adaptation

- **MLLR** model adaptation is performed by $\hat{\mu}_{ik} = M_c \xi_{ik}$ where $\xi_{ik} = [\mu_{ik}^T 1]^T$

- fMLLR feature-space adaptation is performed by $\hat{x}_t = Ax_t + b = M^f t \xi_t$ where $\xi_t = [x_t^T 1]^T$

- Auxiliary function $Q(M^f | M^f(k))$ is formed as

$$- \sum_{i} \sum_{k} \sum_{t=1}^{T} \gamma_{ik}(t) \log |\Sigma_{ik}| - 2 \log |A| + (\hat{x}_t - \mu_{ik})^T \Sigma_{ik}^{-1} (\hat{x}_t - \mu_{ik})$$

- Iterative row-by-row optimization (Gales 1998)
Modern speaker adaptation

• Auxiliary functions for fMLLR and MLLR are jointly optimized with respect to regression parameters $M^f$ and $M$

• fMLLR+MLLR was extended to the adaptation of full covariance matrices $\Sigma_{ik}$ (Povey & Saon 2006)

• Modern LVCSR acoustic models
  – are speaker-adaptively trained in a canonical feature space given by VTLN-warped and fMLLR-transformed features
  – use VTLN, fMLLR and MLLR at test time
Examples of WERs after speaker adaptation

<table>
<thead>
<tr>
<th>Training</th>
<th>WER English CTS 300h</th>
<th>WER English BN 50h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speaker independent</td>
<td>23.3%</td>
<td>30.7%</td>
</tr>
<tr>
<td>VTLN</td>
<td>21.5%</td>
<td>25.8%</td>
</tr>
<tr>
<td>VTLN+fMLLR</td>
<td>19.2%</td>
<td>23.2%</td>
</tr>
<tr>
<td>VTLN+fMLLR+MLLR</td>
<td>18.9%</td>
<td>22.5%</td>
</tr>
</tbody>
</table>
Table of contents

- Introduction
- Front-end processing
- Acoustic modeling
- Deep neural networks
- Language modeling
- Hypothesis search and system combination
- New directions and conclusions
Deep neural networks

- Context-dependent deep neural network HMMs have recently shown significant improvements over discriminatively trained HMMs with state-dependent GMMs (Seide et al. 2011a)

- Deep belief network (DBN) (Hinton et al. 2006) uses a greedy, layer-wise pretraining of the weights with either a supervised or unsupervised criterion

This pretraining step prevents the supervised training of the network from being trapped in a poor local optimum.

- Success of using a deep neural network (DNN) acoustic model in LVCSR has been reported in (Dahl et al. 2012, Hinton et al. 2012).
Hybrid decoding versus feature extraction

Two ways of using MLPs for LVCSR

- “Hybrid” decoding:
  \[
p(x_t|s_t) \propto \frac{p(s_t|x_t)}{p(s_t)}
\]

- Feature extraction:
  - tandem features (Hermansky et al. 2000)
  - bottleneck features (Grezl et al. 2007)
DNN architecture

- Input layer corresponds to spliced fMLLR/fMPE frames
- 5-7 hidden layers with 2048 units and sigmoid non-linearity
- Output layer with 2000-9000 units and softmax non-linearity (1 unit per CD-HMM state)
Cross-entropy training

\[ E = - \sum_i t_i \log y_i, \quad \frac{\partial E}{\partial x_i} = y_i - t_i \]

- Predominant criterion for frame-based classification
- Stochastic gradient descent on mini-batches of 200-500 frames
- Frame randomization (Seide et al. 2011a)
Importance of pretraining

- Results from (Seide et al. 2011b) for a DNN with 7 hidden layers

<table>
<thead>
<tr>
<th>Pretraining</th>
<th>WER English CTS 300h</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>17.4%</td>
</tr>
<tr>
<td>DBN</td>
<td>17.1%</td>
</tr>
<tr>
<td>Layerwise backprop</td>
<td>17.4%</td>
</tr>
<tr>
<td>Discriminative</td>
<td>16.8%</td>
</tr>
</tbody>
</table>

- Other results (courtesy of H. Soltau)

<table>
<thead>
<tr>
<th>Pretraining</th>
<th>WER English BN 50h</th>
<th>WER Levantine RATS 300h</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>17.9%</td>
<td>43%</td>
</tr>
<tr>
<td>Discriminative</td>
<td>17.8%</td>
<td>42%</td>
</tr>
</tbody>
</table>
Sequence-discriminative training

Hessian-free sequence training for DNNs (Kingsbury et al. 2012)

- State-based minimum Bayes risk loss for DNNs

\[
\mathcal{L}(\theta) \triangleq \sum_W p^\kappa(W \mid X) H(W, W^{ref}), \quad \frac{\partial \mathcal{L}}{\partial x_i} = \kappa(\gamma_{i}^{den} - \gamma_{i}^{num})
\]

- Iteratively form quadratic approximation to the loss

\[
\mathcal{L}(\theta + d) \approx \mathcal{L}(\theta) + \nabla \mathcal{L}(\theta)^T d + d^T B(\theta) d
\]

where \( B(\theta) \) is the curvature matrix (damped Gauss-Newton matrix)

- Minimize quadratic approximation with truncated conjugate gradient
# Sequence training results

<table>
<thead>
<tr>
<th>LVCSR task</th>
<th>Cross-entropy</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>English BN 50h SA</td>
<td>15.2%</td>
<td>12.6%</td>
</tr>
<tr>
<td>English CTS 300h SI</td>
<td>16.1%</td>
<td>14.1%</td>
</tr>
<tr>
<td>English CTS 300h SA</td>
<td>14.1%</td>
<td>12.5%</td>
</tr>
<tr>
<td>Levantine RATS 300h SA</td>
<td>42.3%</td>
<td>37.7%</td>
</tr>
</tbody>
</table>
### GMM/DNN comparison

<table>
<thead>
<tr>
<th>LVCSR task</th>
<th>GMM</th>
<th>DNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>English BN 50h SA</td>
<td>14.5%</td>
<td>12.6%</td>
</tr>
<tr>
<td>English CTS 300h SI</td>
<td>18.9%</td>
<td>14.1%</td>
</tr>
<tr>
<td>English CTS 300h SA</td>
<td>15.1%</td>
<td>12.5%</td>
</tr>
<tr>
<td>Levantine RATS 300h SA</td>
<td>46.7%</td>
<td>37.7%</td>
</tr>
</tbody>
</table>
Table of contents

• Introduction
• Front-end processing
• Acoustic modeling
• Deep neural networks
• **Language modeling**
• Hypothesis search and system combination
• New directions and conclusions
Language modeling

- LM plays an important role in LVCSR for finding the best word sequence $\hat{W}$ according to the Bayes decision rule.

- LM based on $n$-gram

\[ p(W) = p(w_1, \cdots, w_T) = \prod_{i=1}^{T} p(w_i|w_1, \cdots, w_{i-1}) \approx \prod_{i=1}^{T} p(w_i|w_{i-n+1}^{i-1}) \]

- $N$-gram probability based on ML is calculated by

\[ p_{ML}(w_i|w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i-1})}{c(w_{i-n+1}^{i-1})} \text{ where } c(w_{i-n+1}^{i-1}) = \sum_{w_i} c(w_{i-n+1}^{i-1}) \]
Language modeling
Issues in LMs

• **Data sparseness** problem - model smoothing
  – backoff method
  – continuous space LM, neural network LM

• **Insufficient long-distance** regularity - topic/class information
  – probabilistic latent semantic analysis (PLSA)
  – latent Dirichlet allocation (LDA)

• **Model regularization** - mismatch between training and test data
  – model “M”
Backoff smoothing

- Chen & Goodman (1999) surveyed a series of smoothing algorithms which cope with zero probability estimates for \( n \)-grams not observed in the training corpus.

- **Interpolation** smoothing is performed by linearly combining higher-order \( n \)-grams with lower-order \( n \)-grams:

\[
p_{\text{INT}}(w_i|w_{i-n+1}^{i-1}) = \lambda_{w_{i-n+1}^{i-1}} p_{\text{ML}}(w_i|w_{i-n+1}^{i-1})
\]

\[
+ (1 - \lambda_{w_{i-n+1}^{i-1}}) p_{\text{INT}}(w_i|w_{i-n+2}^{i-1})
\]

- Parameter \( \lambda_{w_{i-n+1}^{i-1}} \) is estimated for each \( w_{i-n+1}^{i-1} \) by maximum likelihood method.
Other smoothing method

- **Witten-Bell** smoothing (Witten & Bell, 1991)

\[
p_{WB}(w_i|w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^i) + N_{1+}(w_{i-n+1}^{i-1})p_{WB}(w_i|w_{i-n+2}^{i-1})}{c(w_{i-n+1}^{i-1}) + N_{1+}(w_{i-n+1}^{i-1})}
\]

where
\[
N_{1+}(w_{i-n+1}^{i-1}) = |\{w_i : c(w_{i-n+1}^{i-1}w_i) > 0\}|
\]

- **Absolute** discounting

\[
p_{ABS}(w_i|w_{i-n+1}^{i-1}) = \frac{\max\{c(w_{i-n+1}^i) - d, 0\}}{c(w_{i-n+1}^{i-1})} + (1 - \lambda_{w_{i-n+1}^{i-1}})p_{ABS}(w_i|w_{i-n+2}^{i-1})
\]

where 0 ≤ d ≤ 1 denotes a discounting parameter
Kneser-Ney LM

- **Kneser-Ney (KN) smoothing** (Kneser & Ney 1995) conducts language modeling via

\[
p_{\text{KN}}(w_i|w_{i-n+1}^{i-1}) = \frac{\max\{c(w_{i-n+1}^i) - d, 0\}}{c(w_{i-n+1}^{i-1})} \]

\[
+ \frac{d \cdot N_{1+(w_{i-n+1}^{i-1})}}{c(w_{i-n+1}^{i-1})} p_{\text{KN}}(w_i|w_{i-n+2}^{i-1})
\]

or equivalently

\[
p_{\text{KN}}(w_i|w_{i-n+1}^{i-1}) = \begin{cases} 
\frac{\max\{c(w_{i-n+1}^i) - d, 0\}}{c(w_{i-n+1}^{i-1})} & \text{if } c(w_{i-n+1}^i) > 0 \\
\gamma(w_{i-n+1}^{i-1}) p_{\text{KN}}(w_i|w_{i-n+2}^{i-1}) & \text{if } c(w_{i-n+1}^i) = 0
\end{cases}
\]
**Modified Kneser-Ney LM**

- **Modified Kneser-Ney** (MKN) smoothing (Chen 1999) outperformed all other algorithms for LVCSR

\[
p_{\text{MKN}}(w_i|w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i-n}) - d(c(w_{i-n+1}^{i-1}))}{c(w_{i-n+1}^{i-1})} + \gamma(w_{i-n+1}^{i-1})p_{\text{MKN}}(w_i|w_{i-n+2}^{i-1})
\]

where

\[
\gamma(w_{i-n+1}^{i-1}) = \frac{d_1 N_1(w_{i-n+1}^{i-1}) + d_2 N_2(w_{i-n+1}^{i-1}) + d_3 + N_3(w_{i-n+1}^{i-1})}{c(w_{i-n+1}^{i-1})}
\]

\[
d(c) = \begin{cases} 
0 & \text{if } c = 0 \\
d_1 & \text{if } c = 1 \\
d_2 & \text{if } c = 2 \\
d_{3+} & \text{if } c \geq 3 
\end{cases}
\]
Hierarchical Pitman-Yor LM

- HPY-LM is an Bayesian extension of MKN-LM (Teh 2006)
- Gibbs sampling was applied for model inference based on Chinese restaurant metaphor
- HPY-LM improved performance over MKN-LM for LVCSR (Huang & Renals 2010)

\[
p_{\text{HPY}}(w_i|w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^i) - d_{w_{i-n+1}^{i-1}} N_{1+(w_{i-n+1}^i)}}{\theta_{w_{i-n+1}^{i-1}} + c(w_{i-n+1}^{i-1} \cdot)} + \frac{\theta_{w_{i-n+1}^{i-1}} + d_{w_{i-n+1}^{i-1}} N_{1+(w_{i-n+1}^{i-1} \cdot)}}{\theta_{w_{i-n+1}^{i-1}} + c(w_{i-n+1}^{i-1} \cdot)} p_{\text{HPY}}(w_i|w_{i-n+2}^{i-1})
\]
HPY Process

HPY process (Teh 2006) is formed by

\[ G_\phi \sim PY(d_0, \theta_0, G_b) \text{ and } G_U \sim PY(d_{|U|}, \theta_{|U|}, G_{\pi(U)}) \]

\[ G_{\pi(U)} \sim PY(d_{|\pi(U)|}, \theta_{|\pi(U)|}, G_{\pi(\pi(U)))} \]

\[ G_1 \sim PY(d_1, \theta_1, G_{\phi}) \]

\[ G_{\phi} \sim PY(d_0, \theta_0, G_b) \]

\[ p(w_i | w_{i-1}, \ldots, w_{i-n+2}, w_{i-n+1}) \]
Chinese restaurant metaphor

\[
p(\text{occupied table } k|\text{previous customers}) = \frac{c_k - d}{\theta + c}.
\]

\[
p(\text{new table } k|\text{previous customers}) = \frac{\theta + dt}{\theta + c}.
\]

\[
p(z_i = k|z_{-i}) = \begin{cases} 
\frac{c_k - d}{\theta + c}, & 1 \leq k \leq t \\
\frac{\theta + dt}{\theta + c}, & k = t + 1
\end{cases}
\]
Power-law distribution

- Pitman-Yor process (Pitman & Yor 1997) produces power-law distributions which resemble those seen in natural language
  - rich-get-richer property

Figure 1: First panel: number of unique words as a function of the number of words drawn on a log-log scale, with $d = .5$ and $\theta = 1$ (bottom), 10 (middle) and 100 (top). Second panel: same, with $\theta = 10$ and $d = 0$ (bottom), .5 (middle) and .9 (top). Third panel: proportion of words appearing only once, as a function of the number of words drawn, with $d = .5$ and $\theta = 1$ (bottom), 10 (middle), 100 (top). Last panel: same, with $\theta = 10$ and $d = 0$ (bottom), .5 (middle) and .9 (top).

(Teh 2006)
Latent semantic analysis

• **LSA LM** (Bellegarda 2000) was proposed to capture long-range word dependencies through discovery of latent topics

• **Singular value decomposition** is performed over word-document matrix

\[
W \approx U \Sigma V^T
\]

- \( W \) is the word-document matrix
- \( U \) is the word vector
- \( \Sigma \) is a diagonal matrix
- \( V^T \) is the document vector

\( V \times D \) \quad \approx \quad V \times K \quad \Sigma \quad K \times K \quad V^T \times D \)
LSA LM

- LSA LM is calculated from two information sources: \textit{n-grams} and latent topics

\[
p(w_i|h_{i-1}) = p(w_i|h_{i-1}^{(n)}, h_{i-1}^{(l)}) = \frac{p(w_i|h_{i-1}^{(n)})p(h_{i-1}^{(l)}|w_i, h_{i-1}^{(n)})}{\sum_{w_j} p(w_j|h_{i-1}^{(n)})p(h_{i-1}^{(l)}|w_j, h_{i-1}^{(n)})})
\]

\[
= \frac{p(w_i|w_{i-n+1}^{i-1})p(\tilde{d}_{i-1}|w_i)}{\sum_{w_j} p(w_j|w_{i-n+1}^{i-1})p(\tilde{d}_{i-1}|w_j)}
\]

\[
= \frac{p(w_i|w_{i-n+1}^{i-1})p(w_i|\tilde{d}_{i-1})}{\sum_{w_j} p(w_j|w_{i-n+1}^{i-1})p(w_j|\tilde{d}_{i-1})}
\]
PLSA LM

- Topic-based LM (Bellegarda 2000) was proposed to capture long-range word dependencies through discovery of latent topics.

- PLSA LM (Gildea & Hofmann 1999) aims to extract large-span topic information and merge it into LM.

\[
p_{\text{PLSA}}(w_i|w_{i-n+1}^{i-1}) = \sum_{z_i} p(w_i|w_{i-n+1}^{i-1}, z_i) p(z_i|w_{i-n+1}^{i-1})
\]

where \(z_i = k\) denotes topic label of word \(w_i\) determined from history \(h = w_{i-n+1}^{i-1} = \{w_{i-n+1}, \cdots, w_{i-1}\}\).

- Unseen \(n\)-gram events \(p(w_i|w_{i-n+1}^{i-1})\) could not be represented by PLSA.
LDA

- LDA (Blei et al. 2002) estimates the topic model with generalization to unseen test data

\[ p_{\text{LDA}}(w | \alpha, \beta) = \int p(\theta | \alpha) \prod_{n=1}^{N} \sum_{k_n=1}^{K} p(k_n | \theta) p(w_n | k_n, \beta) d\theta \]

- A bag of words \( w = \{w_n\} \) is modeled for document representation
LDA LM

- LDA LM was calculated by using the topic probability estimated
  - from historical words (Tam & Schultz 2005)
  - from transcription of a whole sentence (Tam & Schultz 2006)
LM adaptation

- LDA LM was estimated through LM adaptation using linear interpolation

\[
p_{\text{LDA}}(w_i|w_{i-n+1}^{i-1}) \approx \lambda p_{\text{ngram}}(w_i|w_{i-n+1}^{i-1}) + (1 - \lambda)p_{\text{LDA}}(w_i)
\]

or unigram scaling

\[
p_{\text{LDA}}(w_i|w_{i-n+1}^{i-1}) \approx p_{\text{ngram}}(w_i|w_{i-n+1}^{i-1}) \frac{p_{\text{LDA}}(w_i)}{p_{\text{unigram}}(w_i)}
\]

- Unigram probabilities \(\{p_{\text{LDA}}(w_i)\}\) are trained from a set of documents
Direct model for LM

- **Document**-level topic model (PLSA, LDA)
  - bag-of-words scheme
  - document clustering
  - indirect model

- **$N$-gram**-level class model (DCLM)
  - word order is considered
  - history clustering
  - direct model
History representation

- History words $w_{i-n+1}^{i-1}$ are represented by an $(n-1)V \times 1$ vector

Prediction of topic probabilities $p(c \mid h_{i-n+1}^{i-1})$
(Linear or non-linear classifier)

Prior density of topic mixture

$\theta = [\theta_1, ..., \theta_C]^T \sim \text{Dir}(g(h_{i-n+1}^{i-1}))$
Dirichlet class LM

- **LDA-LM** (Chien & Chueh 2011) is calculated by

\[ p_{DC}(w_i|w_{i-n+1}^i, \mathbf{A}, \mathbf{\beta}) = \sum_{c=1}^{C} \beta_{ic} \frac{g_c(h_{i-n+1}^{i-1})}{\sum_{j=1}^{C} g_j(h_{i-n+1}^{i-1})} \]

![Diagram of Dirichlet class LM](image)
Cache DCLM

- Long-distance topic information is considered to generate topic mixture vector $\theta$

$$p(w_i|h_{i-n+1}^{i-1}, A, \beta, w_1^{i-1}) \approx \sum_{c=1}^{C} \beta_{ic} \frac{g_c(h_{i-n+1}^{i-1}) + \rho \sum_{t=1}^{i-1} \tau_i^{i-t-1} \delta(c, \hat{c}_t)}{\sum_{j=1}^{C} \left[ g_j(h_{i-n+1}^{i-1}) + \rho \sum_{t=1}^{i-1} \tau_i^{i-t-1} \delta(j, \hat{c}_t) \right]}$$
Neural network LM

- **Feedfordward neural network** based LMs outperform standard backoff $n$-gram models (Bengio *et al.* 2003, Schwenk 2007)
  - words are projected into low dimensional space
  - words are automatically clustered together
  - error backpropagation is used for training
  - **multilayer perceptron** (MLP) is performed
NNLM versus DCLM

The $v$-th output: $p(w_i = v | w_{i-n+1}^{i-1})$

Implementation of NNLM
NNLM versus DCLM

The $v$-th output: $p(w_i = v | w_{i-n+1}^{i-1})$

Hierarchical Bayesian modeling

Global projection $g(h_{i-n+1}^{i-1})$

Projection layer

Linear or nonlinear projection

Implementation of DCLM
Comparison of WERs

WSJ0 (87-89 WSJ) and Nov92 ARPA CSR test data were evaluated
Topic-based segmentation

- **Long-term** spoken document transcription
  - segmentation for coherent modeling (Chien & Chueh 2012)
Nonstationary LDA LM

\[
p_{NLDA}(w_i | w_{i-1}^{i-n+1}) \approx \lambda p_{ngram}(w_i | w_{i-n+1}^{i-1}) + (1 - \lambda)p_{NLDA}(w_i)
\]

where

\[
p_{NLDA}(w_i) = \sum_s \sum_{k=1}^{K} \hat{b}_{skw_i} \hat{\gamma}_k \sum_{l=1}^{K} \hat{\gamma}_l
\]

(Chueh & Chien 2009)
Maximum entropy principle

- ME principle (Jaynes 1957) follows two steps
  - **Step 1**: Formulate different information sources as constraints or features
  - **Step 2**: Under these constraints, the distribution with the highest entropy is chosen

- Feature function

\[
    f_k(h, w) = \begin{cases} 
        1, & \text{if } (h, w) \text{ belongs to feature } k \\
        0, & \text{otherwise} 
    \end{cases}
\]

- Constraint

\[
    p(f_k) = \sum_{h, w} p(h, w) f_k(h, w) = \sum_{h, w} \tilde{p}(h, w) f_k(h, w) = \tilde{p}(f_k)
\]
Maximum entropy LM

• **Constrained optimization** is performed by maximizing

\[
\mathcal{F}(p, \lambda) = - \sum_{h, w} \tilde{p}(h)p(w|h) \log p(w|h) + \sum_{k=1}^{F} \lambda_k [p(f_k) - \tilde{p}(f_k)]
\]

• **ME LM** (Rosenfeld 1996) is estimated by

\[
p_{\text{ME}}(w_i|w_{i-n+1}^{i-1}, \lambda) = \frac{\exp \left[ \sum_{k=1}^{F} \lambda_k f_k(W^{r,i}) \right]}{\sum_{w_j} \exp \left[ \sum_{k=1}^{F} \lambda_k f_k(W^{r,j}) \right]}
\]
ME parameters

- **Improved iterative scaling** algorithm (Berger et al. 1996) is used to find ME parameter by

  \[
  \hat{\lambda}_k = \lambda_k + \frac{1}{C_k} \log \frac{E_p[f_k]}{E_{\tilde{p}}[f_k]}, \quad \text{for } k = 1, \ldots, F
  \]

  where \(C_k\) denotes the step size.

- ME LM is proposed to **integrate** different sources including low-order \(n\)-gram, high-order \(n\)-gram, long-distance information and syntactic/semantic knowledge.

- ME technique acts as a **model smoothing** method over different backoff models.
Association patterns in language

(Chien 2006)
### Candidate 1-word subset

<table>
<thead>
<tr>
<th>$C_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taiwan</td>
</tr>
<tr>
<td>Bush</td>
</tr>
<tr>
<td>President</td>
</tr>
<tr>
<td>China</td>
</tr>
<tr>
<td>Alliance</td>
</tr>
<tr>
<td>Palestinian</td>
</tr>
<tr>
<td>Israel</td>
</tr>
<tr>
<td>People</td>
</tr>
<tr>
<td>Army</td>
</tr>
</tbody>
</table>

### Frequent 1-word subset

<table>
<thead>
<tr>
<th>$L_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taiwan</td>
</tr>
<tr>
<td>Bush</td>
</tr>
<tr>
<td>President</td>
</tr>
</tbody>
</table>

### Candidate 2-word subset

<table>
<thead>
<tr>
<th>$C_2$</th>
<th>AMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bush $\cup$ Taiwan</td>
<td>0.22</td>
</tr>
<tr>
<td>President $\cup$ Taiwan</td>
<td>0.16</td>
</tr>
<tr>
<td>Bush $\cup$ President</td>
<td>0.11</td>
</tr>
<tr>
<td>Taiwan $\cup$ President</td>
<td>0.07</td>
</tr>
<tr>
<td>President $\cup$ Bush</td>
<td>0.04</td>
</tr>
<tr>
<td>Taiwan $\cup$ Bush</td>
<td>0.03</td>
</tr>
</tbody>
</table>

### Prune pass

**First Prune Pass**

- **Prune pass**
  - $L_2$
    - Bush $\rightarrow$ Taiwan
    - President $\rightarrow$ Taiwan
    - Bush $\rightarrow$ President

### Join pass

**First Join Pass**

- **Join pass**
  - $C_2$
    - Bush, President $\cup$ Bush
    - Bush, Taiwan $\cup$ President

### Second Prune Pass

- **Second Prune Pass**
  - $\tilde{C}_3$
    - Bush, President $\cup$ Bush
    - AMI 0.12

### Frequent 3-word subset

<table>
<thead>
<tr>
<th>$L_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bush, President $\rightarrow$ Taiwan</td>
</tr>
</tbody>
</table>
Association pattern LM

• Mutual information (MI) in association pattern $W_{d-1}^{r,i} \rightarrow w^r_j$ is merged

$$\log p_{AP}(W) = \sum_{i=1}^{T} \log p(w_i) + \sum_{r=1}^{R} \sum_{w_{d-1}^{r,i} \rightarrow w^r_j \in \Omega_{AP}} \text{MI}(W_{d-1}^{r,i} \rightarrow w^r_j)$$

• Association pattern $L_d = \{W^a_d\} = \{W_{d-1}^{a} \rightarrow w_b\}$ is a frequent $d$-word subset which is referred as a feature in ME model

$$f_{W_{d-1}^{a} \rightarrow w_b}(W^{r,i}) = \begin{cases} 1, & \text{if } W_{d-1}^{a} \in W^{r,i-1}, w_b = w_i \\ 0, & \text{otherwise} \end{cases}$$

$$p_{ME}(w_i | w_{i-n+1}^{i-1}, \lambda) = \frac{\exp \left[ \sum_{k=1}^{F} \lambda_k f_k(W^{r,i}) \right]}{\sum_{w_j} \exp \left[ \sum_{k=1}^{F} \lambda_k f_k(W^{r,j}) \right]}$$
Joint AM & LM

- Joint **acoustic** and **language** (AL) modeling is performed under ME framework (Chien & Chueh 2010)

- **Dependencies** between AM and LM are characterized by **mutual ME** model

\[
p_{\text{AL}}(W|X, \lambda) = \frac{\sum_S p_{\lambda A}(X, S|W)p_{\lambda L}(W)}{\sum_{W'} \sum_S p_{\lambda A}(X|W')p_{\lambda L}(W')} \\
\approx \exp \left[ \sum_{k=1}^{F} \lambda_{k}^{\text{AL}} f_{k}^{\text{AL}}(X, W, \hat{S}) \right] \frac{1}{\sum_{W'} \exp \left[ \sum_{k=1}^{F} \lambda_{k}^{\text{AL}} f_{k}^{\text{AL}}(X, W', \hat{S}) \right]}
\]

- **LM scaling factor** is not required in joint AM & LM
Model regularization

• Chen (2009) addressed the issue of model regularization and found an empirical relationship between cross entropies of training and test data

\[ H_{test} \approx H_{train} + \left( \gamma/N_n \right) \sum_{k=1}^{F} |\tilde{\lambda}_k| \]

– \( N_n \): number of \( n \)-gram events
– \( \tilde{\lambda} = \{\tilde{\lambda}_k\} \): regularized ME parameters
– \( \gamma \): a constant independent of data and model

• This relationship was used to motivate a heuristic for improving LVCSR of test data

• A new middle-sized class-based LM was proposed
Model “M”

- Class-based trigram model is calculated by

\[
p(w_i|w_{i-2}, w_i) = \sum_c p(c_i|w_{i-2}w_{i-1})p(w_i|w_{i-2}w_{i-1}c_i)
\]

where \( p(c_i|w_{i-2}w_{i-1}) \) and \( p(w_i|w_{i-2}w_{i-1}c_i) \) are based on ME model

- **Heuristic 1** Identify groups of features which will tend to have similar \( \lambda_k \) values. For each such feature group, add a new feature to the model that is the sum of the original features

\[
p(y|x, \lambda) = \frac{\exp(3f_1(x, y) + 4f_2(x, y))}{Z(x, \lambda)}
\]

where \( \lambda_1 = 3 \) and \( \lambda_2 = 4 \)
Some heuristics

- A new feature \( f_3(x, y) = f_1(x, y) + f_2(x, y) \) is added. New parameters are set as \( \lambda_1^{\text{new}} = 0 \), \( \lambda_2^{\text{new}} = 1 \) and \( \lambda_3^{\text{new}} = 3 \)

- Model size \( \sum_{k=1}^{F} |\lambda_k| \) is reduced from \( 3 + 4 = 7 \) to \( 0 + 1 + 3 = 4 \). Prediction for test data is improved

- **Heuristic 2** Find a “similar” distribution from an independent training set, and use this distribution as a prior \( q(\cdot) \)

- **Domain adaptation** is applied according to minimum discrimination information (Della Pietra et al. 1992) model

\[
p_{\lambda^{\text{new}}}(y|x) = q(y|x) \frac{\exp(\sum_{k=1}^{F} \lambda_k^{\text{new}} f_k(x, y))}{Z_{\lambda^{\text{new}}}(x)}
\]

Find \( \lambda_{\text{new}} \) such that \( p_{\lambda^{\text{new}}}(y|x) = p_\lambda(y|x) \)
Recurrent neural network LM

- Recurrent verus feedforward neural networks (Mikolov et al. 2010)
  - In feedforward networks, history is represented by context of $n - 1$ words
  - Recurrent networks can learn to compress whole history in low dimensional space, while feedforward networks compress (project) just single word
  - In recurrent networks, history is represented by neurons with recurrent connections - history length is unlimited
Representation in RNNLM

- RNNLM forms an input vector $x_{i-1}$ by concatenating a 1-of-$|V|$ encoded vector $w_{i-1}$ for a history word $w_{i-1}$ and a recurrent state or context vector $s_{i-1}$ from all history words $w_{i-1} = \{w_1, \ldots, w_{i-1}\}$

- State vector $s_i$ is accumulated in all time steps $i$ from history words

- Output layer at neuron $v$ produces the language model $y_{i,v} = p(w_i = v | w_{1}^{i-1})$ for prediction of current word $w_i = v$

- Without the recurrent state vector $s_{i-1}$ and its corresponding weight parameters $R$, RNNLM is simplified as a bigram NNLM
Implementation for RNNLM

Two-layered RNNLM with parameters $\{U, R, L\} = \{\{u_{jk}\}, \{r_{jk}\}, \{l_{kv}\}\}$
RNN unfolded as a **deep feedforward network** - 3 steps back in time
Computation in RNNLM

- Input vector from history words is formed by $x_{i-1} = [w_{i-1}^T \ s_{i-1}^T]^T$

- Hidden neuron $j$ in state vector $s_i$ for current word $w_i$ is obtained by

$$s_{ik} = f \left( \sum_j w_{(i-1)j} u_{jk} + \sum_j s_{(i-1)j} r_{jk} \right)$$

$$f(z) = \frac{1}{1 + e^{-z}}$$

- Output neuron for predicting word $v$ is determined by

$$y_{iv} = p(w_i = v|w_1^{i-1}) = \frac{e^{z_v}}{\sum_{m=1}^{|V|} e^{z_m}}$$

where $g(z_v) = \frac{e^{z_v}}{\sum_{m=1}^{|V|} e^{z_m}}$
Language models using RNN and RNNME

Results on NIST RT04 (Mikolov et al. 2010)
# Table of contents

- Introduction
- Front-end processing
- Acoustic modeling
- Deep neural networks
- Language modeling
- **Hypothesis search and system combination**
- New directions and conclusions
Hypothesis search

- **Weighted finite-state transducers (WFSTs)** compactly encode various knowledge sources
  
  language model, pronunciation dictionary, context decision trees and HMM topologies

- Efficient search with a **time-synchronous Viterbi decoder** (Mohri et al. 2002)
System combination

- Modern LVCSR systems employ multiple decoding and rescoring passes with speaker adaptation passes in between. Performance is improved by cross adaptation.

- ROVER (Fiscus 1998) consists in aligning the word hypotheses from different systems and in outputting the words which have the most votes within each bin.

- Models which are combined usually differ in one or more design parameters such as:
  - input features
  - acoustic modeling paradigm
  - phonetic context
  - discriminative training criterion
Bagging and boosting

- **Bagging** consists in training an ensemble of acoustic models by randomizing the questions in the context **decision trees** (Siohan *et al.* 2005)

- **Boosting iteratively** trains a sequence of acoustic models on re-weighted training samples where the weights of incorrectly decoded frames are increased (Saon & Soltau 2012a)
Comparison between bagging and boosting
### WERs for various LVCSR decoding passes

<table>
<thead>
<tr>
<th>Decoding Pass</th>
<th>WER (English CTS)</th>
<th>WER (Arabic BN)</th>
<th>CER (Mandarin BN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speaker independent</td>
<td>26.7%</td>
<td>16.7%</td>
<td>15.6%</td>
</tr>
<tr>
<td>Speaker adapted</td>
<td>16.4%</td>
<td>8.9%</td>
<td>7.3%</td>
</tr>
<tr>
<td>LM rescoring</td>
<td>15.2%</td>
<td>7.8%</td>
<td>6.5%</td>
</tr>
<tr>
<td>System combination</td>
<td>—</td>
<td>7.4%</td>
<td>6.2%</td>
</tr>
</tbody>
</table>
Table of contents

- Introduction
- Front-end processing
- Acoustic modeling
- Deep neural networks
- Language modeling
- Hypothesis search and system combination
- New directions and conclusions
Subspace Gaussian mixture models

- Povey (2011) presented the SGMMs to allow all phonetic states to share a common GMM structure
  
  - means and mixture weights are varied in a subspace of the entire parameter space
    \[ w_{ijk} = \exp(w_k^T v_{ij}) / \sum_{k'=1}^K \exp(w_{k'}^T v_{ij}) \]
  - each GMM consists of state and sub-state dependent mixture weights
  - mean vectors: \( \mu_{ijk} = \Phi_k v_{ij} \)
  - canonical covariance matrices: \( \Sigma_k \)

\[
\Lambda_{\text{SGMM}} = \{ \Lambda_{ij}, \Lambda_k \} = \{ \{ c_{ij}, v_{ij} \}, \{ \Phi_k, w_k, \Sigma_k \} \}
\]

\[
p_{\text{SGMM}}(x_t | \Lambda_i) = \sum_{j=1}^{N_i} c_{ij} \sum_{k=1}^{K} w_{ijk} N(x_t; \mu_{ijk}, \Sigma_k)
\]
Canonical state models

- State likelihood using CSM (Gales & Yu 2010) is determined in a form of **general transformation** functions by using
  - mixture weights: $w_{ijk} = F_w(k, \theta_{ij})$
  - mean vectors: $\mu_{ijk} = F_\mu(k, \theta_{ij})$
  - covariance matrices: $\Sigma_{ijk} = F_\Sigma(k, \theta_{ij})$

where

- $\theta_{ij}$: the set of transform parameters
- $c_{ij}$: transform prior
- $\Lambda_{ij} = \{c_{ij}, \theta_{ij}\}$

- **CSM** is a general model and can be realized to the mixtures of MLLR transforms, mixtures of fMLLR transforms and SGMMs
Why Bayesian?

Thomas Bayes (1701-1761)

- We are facing the challenges of big data.
- The collected data are prone to be noisy, mislabeled, misaligned, mismatched and ill-posed.
Why Bayesian acoustic model?

- How do we estimate acoustic model from heterogeneous speech data? Is ML-based GMM a right model?

- Are Gaussians over-trained? Too many Gaussians? Are all Gaussians relevant to represent a new speech frame?

- Can we minimize the model assumption error? Can we change model structure?

- How model regularization is assured for unknown test conditions? How model uncertainty is considered? (Bishop 2006)

- Should we collect unlimited speech data for LVCSR?
Basis representation

• Sparse representation of data $x \in \mathcal{R}^D$

\[ x = \Phi w \]

– basis vectors $\Phi = [\phi_1, \cdots, \phi_N]$
– sensing weights $w \in \mathcal{R}^N$
– reconstruction errors $\|x - \Phi w\|_2^2$

• Sensing weights are prone to be sparse especially in representation of ill-posed data
Model construction

- We characterize reconstruction error of an observation $x_t$ by a Gaussian density with zero mean and state-dependent precision matrix $R_i$. State-dependent basis vectors $\Phi = [\phi_1, \cdots, \phi_{iN}]$ are adopted (Saon & Chien 2011, 2012b).

$$p(x_t|\lambda_i) \propto |R_i|^{1/2} \exp \left[ -\frac{1}{2} (x_t - \Phi_i w_t)^T R_i (x_t - \Phi_i w_t) \right]$$

$$= |R_i|^{1/2} \exp \left[ -\frac{1}{2} \left( x_t - \sum_{n=1}^{N} \phi_{in} w_{tn} \right)^T R_i \left( x_t - \sum_{n=1}^{N} \phi_{in} w_{tn} \right) \right]$$

- Point estimates of weight parameters are unreliable
Sparse Bayesian Sensing

- Bayesian sensing aims to yield the error bars or distribution estimates of the true signals

- **Prior** density of sensing weights is incorporated

\[
p(w_t|A_i) = \mathcal{N}(w_t|0, \text{diag}\{\alpha_{in}^{-1}\}) = \prod_{n=1}^{N} \mathcal{N}(w_{tn}|0, \alpha_{in}^{-1})
\]

where \(A_i\) is a state-dependent precision matrix

- Precision parameter \(\alpha_{in}\) is called automatic relevance determination (ARD) which reflects how an observation is relevant to a basis vector
Automatic Relevance Determination

- If ARD is modeled by a gamma density, the marginal distribution of weights turns out to be a Student’s \( t \) distribution which is a sparse prior (Tipping 2001)

\[
p(w_i|a_i, b_i) = \prod_{n=1}^{N} \int_{0}^{\infty} \mathcal{N}(w_{in}|0, \alpha_{in}^{-1}) \mathcal{G}(\alpha_{in}|a_i, b_i) d\alpha_{in}
\]

\[
\propto \prod_{n=1}^{N} (b_i + w_{tn}^2/2)^{-(a_i+1/2)}
\]
BS-HMM parameters

- BS-HMM parameters include
  - basis vectors \( \Phi = [\phi_{i1}, \cdots, \phi_{iN}] \)
  - precision matrix of sensing weights \( A_i \)
  - precision matrix of reconstruction errors \( R_i \)

- Maximum likelihood (ML) type II estimation is performed by

\[
\lambda^{\text{ML}} = \{\pi_i^{\text{ML}}, a_i^{\text{ML}}, A_i^{\text{ML}}, \Phi_i^{\text{ML}}, R_i^{\text{ML}}\} \\
= \arg\max_{\{\pi_i, a_{ki}, A_i, \Phi_i, R_i\}} p(X | \{\pi_i, a_{ki}, A_i, \Phi_i, R_i\})
\]

- EM algorithm is applied for parameter estimation
Some extensions

• Mixture model

\[ p(x_t|\lambda_i) \propto \sum_{k=1}^{K} \omega_{ik}|R_{ik}|^{1/2} \exp \left[ -\frac{1}{2}(x_t - \Phi_{ik}w_t)^TR_{ik}(x_t - \Phi_{ik}w_t) \right] \]

• Basis adaptation for speaker adaptation

\[ \hat{\Phi}_{in} = A\Phi_{in} + b = M\tilde{\Phi}_{in} \quad \hat{\Phi}_i = M\tilde{\Phi}_i \]

• Discriminative training
  – BMMI criterion
  – model space BMMI
  – feature space BMMI
Experimental setup

• **1800 hours** of Arabic broadcast news training data

• VTL-warped PLP cepstra with LDA and STC

• Speaker adaptation with VTLN, FMLLR and multiple MLLR

• Feature and model space discriminative training with BMMI

• Acoustic models have 5000 states and
  – **800K Gaussians** for the baseline
  – **417K Gaussians** for the BS-HMMS

• Recognition vocabulary: 795K words

• Language model: 4-gram with 884M $n$-grams
Model compression using ARD

- Acoustic models built with discriminative feature-space transforms

- Discard 50% of basis vectors corresponding to the largest precision values \( A_i = \{\alpha_{in}\} \) after training

- Results in word error rates before and after discriminative training

<table>
<thead>
<tr>
<th>Model</th>
<th>Training</th>
<th>DEV07</th>
<th>DEV08</th>
<th>DEV09</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>ML type II</td>
<td>12.0%</td>
<td>13.9%</td>
<td>17.4%</td>
</tr>
<tr>
<td>compressed</td>
<td>ML type II</td>
<td>12.4%</td>
<td>14.2%</td>
<td>17.6%</td>
</tr>
<tr>
<td>original</td>
<td>BMMI</td>
<td>10.7%</td>
<td>11.9%</td>
<td>15.0%</td>
</tr>
<tr>
<td>compressed</td>
<td>BMMI</td>
<td>10.4%</td>
<td>11.7%</td>
<td>14.8%</td>
</tr>
</tbody>
</table>
GALE 2011 evaluation results

- All models are cross-adapted on the output of a system using subspace GMMs
- Evaluation test sets: DEV09, EVAL-P4, EVAL-P5
- Word error rates

<table>
<thead>
<tr>
<th>System</th>
<th>DEV09</th>
<th>EVAL-P4</th>
<th>EVAL-P5</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline 800K</td>
<td>13.1%</td>
<td>10.0%</td>
<td>9.4%</td>
</tr>
<tr>
<td>compressed BS-HMM</td>
<td>12.8%</td>
<td>9.7%</td>
<td>9.1%</td>
</tr>
<tr>
<td>system combination</td>
<td>12.6%</td>
<td>9.6%</td>
<td>9.0%</td>
</tr>
</tbody>
</table>
Current trends in DNN acoustic modeling

• Novel architectures
  – Convolutional neural networks (in time and frequency)
  – Recurrent neural networks
  – Bidirectional networks, long-term short-term memory networks

• Speaker adaptation

• Training algorithms for large data
  – Second-order methods (e.g. Hessian-free)
  – GPU implementations
Bayesian Neural Networks

- Bayesian neural networks applied for speech recognition
- Two inference solutions to BNNs.
- The first solution is to calculate the maximum a posteriori estimate of model parameters by considering prior distribution $p(w)$
- MAP estimate is treated as point estimate. This solution can be applied for speaker adaptation
- The second solution is to fulfill full Bayesian where the uncertainty of parameter $w$ is considered for prediction or decoding
- Marginal distribution over $w$ is calculated. Distribution estimate is developed and applied for robust speech recognition
Conclusions

• We have surveyed a series of approaches
  – front-end processing
  – acoustic modeling
  – deep neural networks
  – language modeling
  – hypothesis search
  – system combination

  which have made big contributions for LVCSR

• **Flexible acoustic models** based on **structural state models** and robust basis representation

• Future directions towards **structural learning** and **model regularization** for the different components of an LVCSR system

• Current trends in DNN acoustic modeling
Thanks to

IBM Advanced LVCSR group members

Stephen Chu, Brian Kingsbury, Jeff Kuo, Lidia Mangu, Michael Picheny, Dan Povey, Hagen Soltau and Geoffrey Zweig

& National Science Council of Taiwan

References


Advances in Large Vocabulary Continuous Speech Recognition

Interspeech 2013 tutorial


Contact Information

• George Saon  
  IBM T. J. Watson Research Center  
  gsaon@us.ibm.com  

• Jen-Tzung Chien  
  National Chiao Tung University  
  jtchien@nctu.edu.tw  