Probabilistic Modeling of Speech and Language

Zhijian Ou
Speech Processing and Machine Intelligence (SPMI) Lab, Department of Electronic Engineering, Tsinghua University, Beijing, China. Now Visiting Scholar at Beckman Institute, UIUC.

6/16/2015, IBM
What is this talk about?

• Brief introduction to SPMI lab

• Motivation
• Random field approach to language modeling, ACL 2015.
Overview of SPMI Lab

• Setup the lab, since 2003.
• 2 master and 2 ph.d. students (Current), 7 master students (Graduated).
• Research interests
  • Speech Signal and Information Processing
    • Speech recognition and understanding (LVCSR - Mandarin, English)
    • Source separation
    • Speaker recognition
    • Natural language processing
    • Microphone array
  • Statistical Machine Intelligence
    • Construct probabilistic models of the studied phenomenon using human knowledge and machine learning algorithms;
    • Find efficient ways of implementing probabilistic inference with those models.
Motivation - Probabilistic Modeling of Speech and Language

- **Speech Models**: Speech recognition, pitch estimation, source separation, ...
- **Language Models**: Speech recognition, machine translation, handwriting recognition, ...
- The more scientific the models are, the better we can do for speech and language processing.

\[
\arg\max_W P(W|X) = \arg\max_W \frac{P(X|W)P(W)}{P(X)}
\]
What is this talk about?

• Brief introduction to SPMI lab

• Motivation
  • Random field approach to language modeling, ACL 2015.
Our trial-and-error efforts

• Relax the state independent assumption in HMMs

• Bayesian HMM modeling of speech
  • ICASSP 2007

• Variational nonparametric Bayesian HMM
  • ICASSP 2010

• NMF modeling of voice in song, and a monaural voice and accompaniment separation system
  • ICASSP 2011.

• Eigenvoice Speaker Modeling + VTS-based Environment Compensation for Robust Speech Recognition
  • ICASSP 2012

• PAT Models
  • AISTATS 2012, ICASSP 2014
Motivation

What is the basic physical model of speech production?

— The Acoustic Tube Model, a.k.a Source-Filter Model.
Motivation

Are there any generative models of speech?
Motivation

• Most of them are actually generative models of the speech features
  • e.g. Magnitude, Cepstrum, Correlogram

• Only a few directly model the spectrogram
  • Reyes-Gomez, Jojic, Ellis, 2005; Bach and Jordan, 2005; Kameoka et al. 2010; Hershey et al. 2010; Deng et al. 2006.

• None of them fully respect the physical acoustic tube model

  Important speech elements
  • Pitch
  • Glottal source
  • Vocal tract response
  • Aspiration noise
  • Phase
Motivation

• **Drawback:** Speech analysis is inaccurate, making great troubles for back-end inference
  • Chicken and egg effect \(^1\)
  • Entangled variation/randomness
  • e.g. Vocal tract estimate (e.g. LPC and MFCC) corrupted by ‘spectral tilt’ due to glottal pulse

• **A complete model of speech**
  • Disentangle the underlying elements of variation, knowledgeably vs blindly.
  • Provide strong constraints/priori knowledge \(^2\)


Motivation

• Previous efforts
  • Additive deterministic-stochastic model, (Serra & Smith 1990)
  • STRAIGHT model, (Kawahara, et al. 2008)
  • Mixed source model and its adapted vocal tract filter estimate for voice transformation and synthesis, (Degottex, et. al 2013)
  • Non-negative source-filter dynamical system for speech enhancement, (Simsekli, Le Roux, Hershey, 2014)

• Probabilistic Acoustic Tube (PAT)
  • Jointly consider breathiness, glottal excitation and vocal tract in a probabilistic modeling framework, and notably with phase information.


PAT2: Improvement of PAT Model for Speech Decomposition. (Zhang, Ou, Hasegawa-Johnson. ICASSP 2014)

PAT3: Incorporating AM-FM effect in voiced speech for PAT model. (Zhang, Ou, Hasegawa-Johnson. Submitted)
PAT2 Model

Doval et al 2013

\[ e_v[t] = \sum_d \text{real}[G(d\omega_0) \cdot e^{id\omega_0(t-\tau)}] \]

\[ s[t] = v[t] + u[t] \]
\[ = (a \cdot e_v[t] + b \cdot e_u[t]) \ast h[t] \]

26-dim Complex Cepstrum \( \hat{h} \) with quefrency \( \hat{t} \)

Impulse response of vocal tract

Serra & Smith 1990, Degottex et al 2013

Voiced speech

Unvoiced speech

\[ e_u[t] \sim \mathcal{N}(0,1), \text{i.e. WGN} \]
Three-pole Model for Glottal Pulse (Doval et al 2013)

Glottal Flow Waveform

\[ e_v[t] = \sum_d \text{real}[G(d\omega_0) \cdot e^{j\omega_0(t-\tau)}] \]

\[ G(\omega) = \frac{1}{[1 + 2g_1 \cos(\beta)e^{-j\omega} + g_1^2 e^{-2j\omega}][1 + g_2 e^{-j\omega}]} \]

parameterized by \( \tilde{g} = \{g_1, \beta, g_2\} \)
PAT2 Summary

**Time domain:**
\[ s[t] = v[t] + u[t] = (a \cdot e_v [t] + b \cdot e_u [t]) \times h[t] \]

**Frequency domain:**
\[ \tilde{s} = a \cdot vec(\omega_0, \tau, \tilde{g}, \hat{h}) + b \cdot vec[DFT[h[t]]] \odot vec[DFT[WGN]] \]

**Hidden variables:**
\[ z = \{a, b, \omega_0, \tau, \tilde{g}, \hat{h}\} \in R^{31} \]

**MAP inference**
\[ p(z|\tilde{s}) \propto p(\tilde{s}|z)p(z) \] by Monte Carlo sampling and L-BFGS search.
### Experimental Results

<table>
<thead>
<tr>
<th>Voiced Reconstruction</th>
<th>Voiced Reconstruction – Single Frame</th>
<th>GCI Location Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Spectrogram</td>
<td>Real Spectrum</td>
<td></td>
</tr>
<tr>
<td>Voiced Reconstruct</td>
<td>Imaginary Spectrum</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vocal Tract Filter Estimation</th>
<th>Voiced vs Whispered</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAT2</td>
<td>PAT2</td>
</tr>
<tr>
<td>MFCC</td>
<td>MFCC</td>
</tr>
</tbody>
</table>
PAT3 Motivation

• To incorporate AM-FM effect in voiced speech
  • Harmonic part is assumed to be strictly periodic.
  • Variations within a single voiced frame are common and non-negligible.

• Two main variations are pitch jitter and amplitude shimmer
  • Give voiced speech its naturalness
PAT2 Model

\[ v[t] = \sum_d \text{real}[\alpha_d e^{jd\omega_0 t}] \]

where \( \alpha_d = aH(d\omega_0)G(d\omega_0)e^{-jd\omega_0 \tau} \)

\[ v[t] = \sum_d x_d[t]^T \xi_d[t] \]

\[ x_d[t] = \begin{pmatrix} |\alpha_d| \cos(d\omega_0 t + \angle \alpha_d) \\ |\alpha_d| \sin(d\omega_0 t + \angle \alpha_d) \end{pmatrix}, \text{the strictly periodic signal} \]

\[ \xi_d[t] = \begin{pmatrix} \eta_d[t] \cos(d\phi[t]) \\ \eta_d[t] \sin(d\phi[t]) \end{pmatrix}, \text{the amplitude and phase perturbation, phasor} \]

PAT3 Model

\[ v[t] = \sum_d \text{real}[\alpha_d \eta_d[t] e^{jd\omega_0 t + jd\phi[t]}] \]

Amplitude perturbation

Phase perturbation
An AM-FM sinusoid

\[ y(t) = \Re(a(t) \exp(i\phi(t))) \]
Phasor representation

\[ y(t) = \Re (a(t) \exp(i\phi(t))) \]
Phasor representation

\[ y(t) = \Re(a(t) \exp(i\phi(t))) \]
Phasor representation

\[ y(t) = \Re (a(t) \exp(i\phi(t))) \]
If $\phi(t) = \omega t + \theta(t)$, then

$$y(t) = \text{real}[a(t)e^{j\omega t + j\theta(t)}]$$

$$= f(t)^T \xi(t)$$

$$f(t) = \begin{pmatrix} \cos(\omega t) \\ \sin(\omega t) \end{pmatrix}, \text{ a fixed freq signal}$$

$$\xi(t) = \begin{pmatrix} a(t)\cos(\theta(t)) \\ a(t)\sin(\theta(t)) \end{pmatrix}, \text{ a phasor}$$

**Theorem:** If $\theta(t)$ is uniform distributed, $a(t)$ is Rayleigh distributed,

Then $\xi(t) \sim \mathcal{N} \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \sigma^2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right)$
**PAT3 Model**

\[ v[t] = \sum_d x_d[t]^T \xi_d[t] \]

\[ \begin{cases} x_d[t] = \begin{pmatrix} |\alpha_d| \cos(d\omega_0 t + \angle \alpha_d) \\ |\alpha_d| \sin(d\omega_0 t + \angle \alpha_d) \end{pmatrix}, & \text{the strictly periodic signal} \\ \\ \xi_d[t] = \begin{pmatrix} \eta_d[t] \cos(d\phi[t]) \\ \eta_d[t] \sin(d\phi[t]) \end{pmatrix}, & \text{the amp. & phase perturbation, phasor} \end{cases} \]

\[ \xi_d(t) \sim \mathcal{N} \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \sigma_d^2 \begin{pmatrix} 1 & 0 \\ 0 & \rho_d \end{pmatrix} \right) \]

It can be shown that \[
\begin{align*}
\sigma_d &= \frac{c}{\sqrt{1-e^{-2d\cdot\delta}}} \\
\rho_d &= \tanh(2 \cdot d \cdot \gamma)
\end{align*}
\]

Time domain:
\[ s[t] = v[t] + u[t] = v[t] + (b \cdot e_u[t]) \ast h[t] \]

Frequency domain:
\[ \tilde{s} = \text{vec}(\omega_0, \tau, \tilde{g}, \tilde{h}; \delta, \gamma) + b \cdot \text{vec} \left[ \text{DFT}[h[t]] \right] \sqcap \text{vec}[\text{DFT}[\text{WGN}]] \]

Hidden variables:
\[ z = \{a, b, \omega_0, \tau, \tilde{g}, \tilde{h}; \delta, \gamma\} \in \mathbb{R}^{31+2} \]

MAP inference \[ p(z|\tilde{s}) \propto p(\tilde{s}|z)p(z) \] by Monte Carlo sampling and L-BFGS search.
Experiment - Reconstruction of Voiced Speech with Heavy AM/FM Effect

AM-FM widens the harmonic pulses
Under-estimate of voiced energy 😞
PAT – Summary

• One of the reviewers comments "to my knowledge the most complete attempt on developing a true generative model for speech".

UTML TR 2006–004

To Recognize Shapes, First
Learn to Generate Images

Geoffrey Hinton
Department of Computer Science, University of
Toronto
PAT – Future work

- **PAT: On the way ...**
  - A sequential inference algorithm for nonlinear state-space model
  - Large scale experiments
What is this talk about?

• Brief introduction to SPMI lab

• Motivation
  • Random field approach to language modeling, ACL 2015.
Content

Random Field Language Models (RFLMs) – brand new

• State-of-the-art LMs - review
  • N-gram LMs
  • Neural network LMs

• Motivation - why

• Model formulation - what

• Model Training - breakthrough

• Experiment results - evaluation

• Summary
N-gram LMs

• Language modeling (LM) is to determine the joint probability of a sentence, i.e. a word sequence.

• Dominant: Conditional approach

\[
p(x_1, x_2, \ldots, x_l) = \prod_{i=1}^{l} p(x_i|x_1, \ldots, x_{i-1})
\]

\[
\approx \prod_{i=1}^{l} p(x_i|x_{i-n+1}, \ldots, x_{i-1})
\]

• Using Markov assumption leads to the N-gram LMs
  – One of the state-of-the-art LMs
Neural network LMs

• Another state-of-the-art LMs

\[ p(x_i | x_1, \ldots, x_{i-1}) \approx p(x_i | \phi(x_1, \ldots, x_{i-1})) \]

\[ p(x_i = k | x_1, \ldots, x_{i-1}) \approx \frac{\phi^T w_k}{\sum_{k=1}^{V} \phi^T w_k} \]

where \( V \) is lexicon size, \( w_k \in R^h \)

泅 Computational very expensive in both training and testing \(^1\)

e.g. \( V = 10k \sim 100k, h = 250 \)

\(^1\) Partly alleviated by using un-normalized models, e.g. through noise contrastive estimation training.
RFLMs – Motivation (1)

\[ p(x_1, x_2, \ldots, x_l) = ? \]

Dominant:
Conditional approach / Directed

Alternative:
Random field approach / Undirected

ם Difficulty in model training
ם A rule in language cognition: employ context for reading and writing
The cat is on the table.
The cat is in the house.

ם Breakthrough in training with a number of innovations
Fixed-dimensional (e.g. image) -> Trans-dimensional (sequential modeling)
RFLMs – Motivation (2)

• **Drawback of N-gram LMs**
  - N-gram is only one type of linguistic feature/property/constraint
  - meeting on Monday
    \[ P(w_i = \text{Monday} | w_{i-2} = \text{meeing}, w_{i-1} = \text{on}) \]

• What if the training data only contain ‘meeting on Monday’?
  - New feature ‘meeting on DAY-OF-WEEK’, using class
  - New feature ‘party on *** birthday’, using skip
  - New features ....

• Jelinek 1995: put language back into language modeling
RFLMs – Formulation

• Intuitive idea
  • Features \((f_i, i = 1, 2, \ldots, F)\) can be defined arbitrarily, beyond the n-gram features.
  • Each feature brings a contribution to the sentence probability \(p(x)\)

• Formulation

\[
p(x) = \frac{1}{Z} \exp \left( \sum_{i=1}^{F} \lambda_i f_i(x) \right), x \triangleq (x_1, x_2, \ldots, x_l)
\]

\[
f_i(x) = \begin{cases} 
1, & \text{`meeting on DAY-OF-WEEK' appears in } x \Rightarrow \lambda_i \text{ is activated} \\
0, & \text{Otherwise} \Rightarrow \lambda_i \text{ is removed}
\end{cases}
\]

😊 More flexible features, beyond the n-gram features, can be well supported in RFLMs.
😊 Computational very efficient in computing sentence probability.
Introduction

• Whole-sentence maximum entropy (WSME)
  \[
p(x; \lambda) = \frac{1}{Z(\lambda)} \exp[\lambda^T f(x)]
\]

• The empirical results of previous WSME models are not satisfactory
  • After incorporating lexical and syntactic information, 1% and 0.4% respectively in perplexity and in WER is reported for the resulting WSEM (Rosenfeld et al., 2001).
WSME – Difficulty in model training

\[ p(x; \lambda) = \frac{1}{Z(\lambda)} \exp[\lambda^T f(x)] \]

Normalization constant:

\[ Z(\lambda) = \sum_x \exp \left( \sum_{i=1}^{F} \lambda_i f_i(x) \right) \]

- Maximum-likelihood training

\[ \frac{\partial \text{LogLikelihood}}{\partial \lambda} = E_{\tilde{p}(x)}[f_i(x)] - E_{p(x;\lambda)}[f_i(x)] = 0 \]

- Expectation under empirical distribution \( \tilde{p}(x) \)
- Expectation under model distribution \( p(x;\lambda) \)
RFLMs vs WSME

• Whole-sentence maximum entropy (WSME)

\[ p(l, x^l; \lambda) = \frac{1}{Z(\lambda)} \exp[\lambda^T f(x^l)], \quad x = (l, x^l), \quad x^l \triangleq (x_1, x_2, \ldots, x_l) \]

Essentially a mixture distribution with unknown weights (differ from each other greatly, \(10^{40}\))!
Poor sampling \(\rightarrow\) poor estimate of gradient \(\rightarrow\) poor fitting

\[ p(l, x^l; \lambda) = \frac{Z_l(\lambda)}{Z(\lambda)} \cdot \frac{1}{Z_l(\lambda)} \cdot \exp[\lambda^T f(x^l)], \quad Z_l(\lambda) = \sum_{x^l} \exp[\lambda^T f(x^l)] \]
RFLMs vs WSME

• Whole-sentence maximum entropy (WSME)

\[ p(l, x^l; \lambda) = \frac{1}{Z(\lambda)} \exp[\lambda^T f(x^l)], \quad x \triangleq (l, x^l), \quad x^l \triangleq (x_1, x_2, \ldots, x_l) \]

Essentially a mixture distribution with unknown weights (differ from each other greatly, \(10^{40}\)) ! Poor sampling \(\rightarrow\) poor estimate of gradient \(\rightarrow\) poor fitting

\[ p(l, x^l; \lambda) = \frac{Z_l(\lambda)}{Z(\lambda)} \cdot \frac{1}{Z_l(\lambda)} \cdot \exp[\lambda^T f(x^l)], \quad Z_l(\lambda) = \sum_{x^l} \exp[\lambda^T f(x^l)] \]

• We propose a trans-dimensional RF model

\[ p(l, x^l; \lambda) = \pi_l \cdot \frac{1}{Z_l(\lambda)} \cdot \exp[\lambda^T f(x^l)], \quad l = 1, \ldots, m \]

Empirical length probabilities in the training data
Serve as a control device to improve sampling from multiple distributions !
Introduction to Stochastic Approximation (SA)

Problem: The objective is to find a solution $\theta$ to $E_{Y \sim f(\cdot; \theta)}[H(Y; \theta)] = \alpha$, where $\theta \in R^d$, noisy observation $H(Y; \theta) \in R^d$

Method:

1. Generate $Y_t \sim K(Y_{t-1}, \cdot; \theta_{t-1})$, a Markov transition kernel that admits $f(\cdot; \theta_{t-1})$ as the invariant distribution.
2. Set $\theta_t = \theta_{t-1} + \gamma_t \{\alpha - H(Y_t; \theta_{t-1})\}$

E.g. $\gamma_t = \frac{1}{t_0 + t}$

Apply SA to RFLM training

- The trans-dimensional RF model

\[
p(l, x^l; \lambda) = \pi_l \cdot \frac{1}{Z_l(\lambda)} \cdot \exp[\lambda^T f(x^l)]
\]

\[
E_{\tilde{p}(x)}[f_i(x)] - E_{p(x; \lambda)}[f_i(x)] = 0, \quad x \triangleq (l, x^l)
\]

- Consider the joint distribution of the pair \((l, x^l)\)

\[
p(l, x^l; \lambda, \zeta) \propto \pi_l \cdot \frac{1}{e^{\zeta_l}} \cdot \exp[\lambda^T f(x^l)]
\]

where \(\zeta_l\) is hypothesized values of the true \(\zeta_l^*(\lambda) = \log Z_l(\lambda)\).

The marginal probability of length \(l\) is:

\[
p(l; \lambda, \zeta) = \frac{\pi_le^{-\zeta_l+\zeta_l^*(\lambda)}}{\sum_j\pi_je^{-\zeta_j+\zeta_j^*(\lambda)}}.
\]

- SA is used to find \(\zeta_l^* = \zeta_l^*(\lambda^*)\) and \(\lambda^*\) that solves

\[
\begin{bmatrix}
\pi_l = p(l; \lambda, \zeta), \quad l = 1, \ldots, m \\
0 = E_{\tilde{p}(x)}[f_i(x)] - E_{p(l, x^l; \lambda, \zeta)}[f_i(x)]
\end{bmatrix}
\]
RFLMs – Breakthrough in training (1)

• Propose Joint Stochastic Approximation (SA) Training Algorithm
  • Simultaneously updates the model parameters and normalization constants

Algorithm 1 Joint stochastic approximation

\begin{itemize}
  \item Input: training set
  \item 1: set initial values \( \lambda^{(0)} = (0, \ldots, 0)^T \) and \( \zeta^{(0)} = \zeta^*(\lambda^{(0)}) - \zeta^{*1}(\lambda^{(0)}) \)
  \item 2: for \( t = 1, 2, \ldots, t_{\text{max}} \) do
  \item 3: \( B^{(t)} = \emptyset \)
  \item 4: set \( (L^{(t,0)}, X^{(t,0)}) = (L^{(t-1,K)}, X^{(t-1,K)}) \)
  \item \textbf{Step I: MCMC sampling}
  \item 5: for \( k = 1 \rightarrow K \) do
  \item 6: sampling (See Algorithm 3)
  \item \( (L^{(t,k)}, X^{(t,k)}) = \text{SAMPLE}(L^{(t,k-1)}, X^{(t,k-1)}) \)
  \item 7: set \( B^{(t)} = B^{(t)} \cup \{(L^{(t,k)}, X^{(t,k)})\} \)
  \item \textbf{Step II: SA updating}
  \item 8: Compute \( \lambda^{(t)} \) based on (13)
  \item 9: Compute \( \zeta^{(t)} \) based on (14) and (15)
  \item 10: end for
\end{itemize}
RFLMs – Breakthrough in training (2)

- Propose Trans-dimensional mixture sampling
  - Sampling from \( p(l, x^t; \lambda, \zeta) \), a mixture of RFs on subspaces of different dimensions.
  - Formally like RJ-MCMC.
RFLMs – Breakthrough in training (3)

- Exploit Hessian diagonal in SA
- Introduce training set mini-batching

Improve the convergence!
Random Field Language Models (RFLMs) – brand new

- State-of-the-art LMs - review
  - N-gram LMs
  - Neural network LMs
- Motivation - why
- Model formulation - what
- Model Training - breakthrough
- Experiment results - evaluation
- Summary
Experiment setup

• **LM Training — Penn Treebank portion of WSJ corpus**
  - Vocabulary: 10K words
  - Training data: 887K words, 42K sentences
  - Development data: 70K words
  - Testing data: 82K words

• **Test speech — WSJ’92 set (330 sentences)**
  - By rescoring of 1000-best lists

• **Various LMs**
  - KN4 (Kneser-Ney)
    - 4gram LMs with modified Kneser-Ney smoothing
  - RNNLMS (Recurrent Neural Network LMs)
    - Trained by the RNNLM toolkit of Mikolov
    - The dimension of hidden layer = 250. Mini-batch size=10, learning rate=0.1, BPTT steps=5.
    - 17 sweeps are performed before stopping (takes about 25 hours). No word classing is used.
  - RFLMs
    - A variety of features based on word and class information
Feature Definition

<table>
<thead>
<tr>
<th>Type</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>w</td>
<td>((w_{-3}w_{-2}w_{-1}w_0)(w_{-2}w_{-1}w_0)(w_{-1}w_0)(w_0))</td>
</tr>
<tr>
<td>c</td>
<td>((c_{-3}c_{-2}c_{-1}c_0)(c_{-2}c_{-1}c_0)(c_{-1}c_0)(c_0))</td>
</tr>
<tr>
<td>ws</td>
<td>((w_{-3}w_0)(w_{-3}w_{-2}w_0)(w_{-3}w_{-1}w_0)(w_{-2}w_0))</td>
</tr>
<tr>
<td>cs</td>
<td>((c_{-3}c_0)(c_{-3}c_{-2}c_0)(c_{-3}c_{-1}c_0)(c_{-2}c_0))</td>
</tr>
<tr>
<td>wsh</td>
<td>((w_{-4}w_0)(w_{-5}w_0))</td>
</tr>
<tr>
<td>csh</td>
<td>((c_{-4}c_0)(c_{-5}c_0))</td>
</tr>
<tr>
<td>cpw</td>
<td>((c_{-3}c_{-2}c_{-1}w_0)(c_{-2}c_{-1}w_0)(c_{-1}w_0))</td>
</tr>
</tbody>
</table>

\(w / c\) : the word/class ngram features up to order 4
\(ws / cs\) : the word/class skipping ngram features up to order 4
\(wsh / csh\) : the higher-order word/class features
\(cpw\) : the crossing class-predict-word features up to order 4
Word Error Rate (WER) results for speech recognition

<table>
<thead>
<tr>
<th>model</th>
<th>WER</th>
<th>PPL (± std. dev.)</th>
<th>#feat</th>
</tr>
</thead>
<tbody>
<tr>
<td>KN4</td>
<td>8.71</td>
<td>295.41</td>
<td>1.6M</td>
</tr>
<tr>
<td>RNN</td>
<td>7.96</td>
<td>256.15</td>
<td>5.1M</td>
</tr>
</tbody>
</table>

RFLMs (100c)

<table>
<thead>
<tr>
<th>model</th>
<th>WER</th>
<th>PPL (± std. dev.)</th>
<th>#feat</th>
</tr>
</thead>
<tbody>
<tr>
<td>w+c</td>
<td>8.56</td>
<td>268.25±3.52</td>
<td>2.2M</td>
</tr>
<tr>
<td>w+c+ws+cs</td>
<td>8.16</td>
<td>265.81±4.30</td>
<td>4.5M</td>
</tr>
<tr>
<td>w+c+ws+cs+cpw</td>
<td>8.05</td>
<td>265.63±7.93</td>
<td>5.6M</td>
</tr>
<tr>
<td>w+c+ws+cs+wsh+csh</td>
<td>8.03</td>
<td>276.90±5.00</td>
<td>5.2M</td>
</tr>
</tbody>
</table>

RFLMs (200c)

<table>
<thead>
<tr>
<th>model</th>
<th>WER</th>
<th>PPL (± std. dev.)</th>
<th>#feat</th>
</tr>
</thead>
<tbody>
<tr>
<td>w+c</td>
<td>8.46</td>
<td>257.78±3.13</td>
<td>2.5M</td>
</tr>
<tr>
<td>w+c+ws+cs</td>
<td>8.05</td>
<td>257.80±4.29</td>
<td>5.2M</td>
</tr>
<tr>
<td>w+c+ws+cs+cpw</td>
<td>7.92</td>
<td>264.86±8.55</td>
<td>6.4M</td>
</tr>
<tr>
<td>w+c+ws+cs+wsh+csh</td>
<td>7.94</td>
<td>266.42±7.48</td>
<td>5.9M</td>
</tr>
</tbody>
</table>

RFLMs (500c)

<table>
<thead>
<tr>
<th>model</th>
<th>WER</th>
<th>PPL (± std. dev.)</th>
<th>#feat</th>
</tr>
</thead>
<tbody>
<tr>
<td>w+c</td>
<td>8.72</td>
<td>261.02±2.94</td>
<td>2.8M</td>
</tr>
<tr>
<td>w+c+ws+cs</td>
<td>8.29</td>
<td>266.34±6.13</td>
<td>5.9M</td>
</tr>
</tbody>
</table>

Table 3: The WERs and PPLs on the WSJ’92 test data. “#feat” denotes the feature number. Different RFLMs with class number 100/200/500 are reported (denoted by “100c”/“200c”/“500c”)

- **Encouraging performance**
  - The RFLM using the “w+c+ws+cs+cpw” features with class number 200 performs comparable to the RNNLM, but is computationally more efficient in computing sentence probability.
  
  Re-ranking of the 1000-best list for a sentence takes 0.16 sec. vs 40 sec.

- **Efficient in training**
  - The WER relative reduction is **9.1%** compared with the KN4, and **0.5%** compared with the RNNLM.

  - Training the RFLM with up to **6 million** features, takes 15 hours.
Summary

Contribution

• Breakthrough in training with a number of innovations.
• Successfully train RFLMs and make performance improvements.

<table>
<thead>
<tr>
<th></th>
<th>Computation efficient in training</th>
<th>Computation efficient in test</th>
<th>Bidirectional context</th>
<th>Flexible features</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-gram LMs</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Neural network LMs</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>RFLMs</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

Future work

• Train RFLMs with richer features on larger-scale corpus.
• Features selection strategy such as L1 regularization.
Thanks:
Yang Zhang, Bin Wang, Mark Hasegawa-Johnson, Zhiqiang Tan.

Thanks for your attention!