

# STRING AND LATTICE BASED DISCRIMINATIVE TRAINING FOR THE CORPUS OF SPONTANEOUS JAPANESE LECTURE TRANSCRIPTION TASK

Erik McDermott & Atsushi Nakamura, *NTT Communication Science Laboratories, NTT Corporation.*

## Previous work:

- (McDermott et al., ASLP Transactions 2007)
- LM = 68,000 word unigram
- Diagonal covariance HMMs
- String-based MCE training

## Here:

- LM = 100,000 word N-gram
- Diagonal & full covariance HMMs
- Both string & lattice based MCE training
  - Numerical subtraction of reference lattice from competitor lattice (use of Macherey et al. (2005) proposal)

## Corpus of Spontaneous Japanese

- Lecture speech transcription task (Maekawa et al., 2000)
- Training set: A set (male + female), i.e. 190,000 utterances (230 hours)
- Test set: standard set of 10 lecture speeches (130 minutes)
- LM during testing: 100,000 word trigram (Kneser-Ney smoothing)
- LM data: text for 2672 lectures in CSJ database
  - Word units occurring in lecture transcriptions contain:
    1. kanji/kana written form
    2. phonetic transcription
    3. part of speech annotation
 → word entry = part1+part2+part3
- Word segmentations exist only at lecture level
- Utterance transcriptions use *phrase-based* segmentation with only kanji/kana written forms

→ Discriminative Training on CSJ: to generate utterance references that use *same* word units as recognition LM, phrase data was *aligned to lecture texts* and corresponding 3 part word units used.

## Discriminative training for string sets

- Overall loss function:

$$\mathcal{F}(\Lambda, \mathcal{X}) = \sum_r f(\underbrace{g_C(\mathcal{X}_r, \Lambda)}_{\text{competitor}}) - \underbrace{g_R(\mathcal{X}_r, \Lambda)}_{\text{reference}}$$

- MCE: reference  $\subset$  competitor set & arbitrary  $f()$
- MMI: reference  $\subset$  competitor set & linear  $f()$
- Discriminant function for string set  $\mathcal{J}$ :

$$g_{\mathcal{J}}(\mathcal{X}_r, \Lambda) = \frac{1}{\psi} \log \left[ \sum_{j|S_j \in \mathcal{J}} e^{g_j(\mathcal{X}_r, \Lambda)\psi} \right]$$

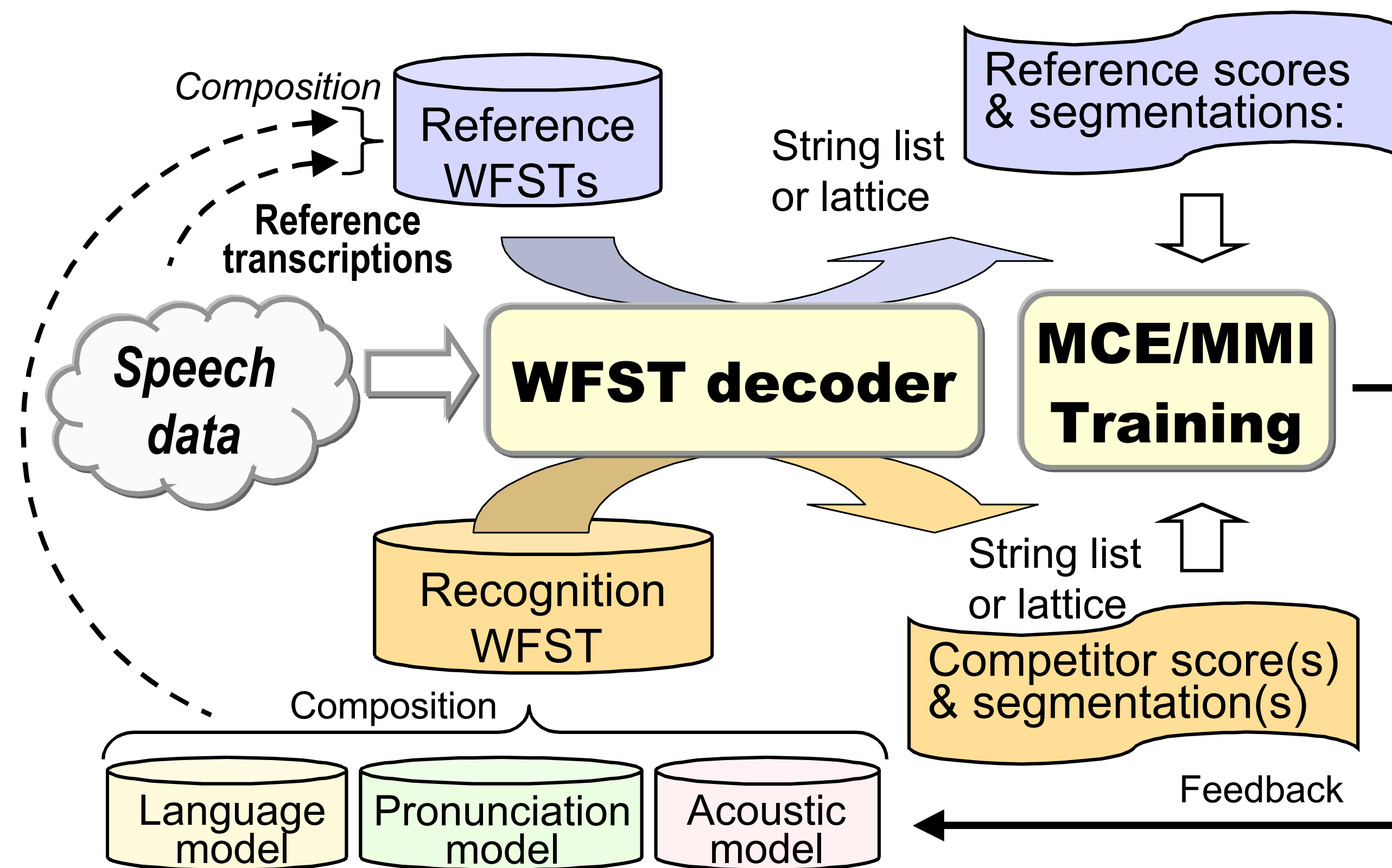
- “Mini”-discriminant function for specific string  $j$ :
 
$$g_j(\mathcal{X}_r, \Lambda) = \eta \log P(S_j) + \log p_{\Lambda}(\mathcal{X}_r|S_j)$$
- Derivative of string set function w.r.t. specific string:

$$\frac{\partial g_{\mathcal{J}}}{\partial g_j} = \frac{P(S_j)^{\eta\psi} p_{\Lambda}(\mathcal{X}_r|S_j)^{\psi}}{\sum_{S_i \in \mathcal{J}} P(S_i)^{\eta\psi} p_{\Lambda}(\mathcal{X}_r|S_i)^{\psi}} \quad (i)$$

- Parallelized gradient computation → Optimization using RPROP algorithm (Leroux, 2005)

## WFST-based training scheme

- String sets represented as Weighted Finite State Transducers (Mori et al. 2000)
- WFST decoder (Hori et al. 2004) used to find best string list or lattice within string set, with corresponding time information



## Training LM = 100,000 word N-gram

- 100,000 word trigram used for both training and testing
- String-level training:
  - Decoder finds top string(s) within reference/competitor set
  - Explicit calculation of Equ. (1)
- Test results (WER), MCE vs. Maximum Likelihood (ML) baseline:

| States | Gaussians | ML   | MCE         |
|--------|-----------|------|-------------|
| 2000   | 16        | 23.0 | 20.2 (12.2) |
| 3000   | 16        | 22.4 | 20.5 (8.5)  |
| 4000   | 16        | 22.1 | 20.3 (8.1)  |
| 5000   | 32        | 21.6 | 19.8 (8.3)  |

(cf MCE unigram results, WER in 20.1% - 21.1% range)

## Full covariance HMMs

- Significant gains on CSJ for ML baseline using full (untied) covariance Gaussians
  - Cholesky decomposition
  - Heavier but still reasonable computation during recognition
- Investigated gains for discriminative training with full covariances
  - String-level training, 100,000 word N-gram
  - Test results (WER):

| States | Gaussians | ML   | MCE        |
|--------|-----------|------|------------|
| 2000   | 8         | 20.2 | 19.4 (4.0) |
| 2000   | 16        | 19.8 | 18.9 (4.6) |
| 3000   | 8         | 20.1 | 19.0 (5.5) |
| 3000   | 16        | 19.4 | 18.9 (2.6) |
| 3000   | 32        | 20.7 | -          |

## Lattice-based training

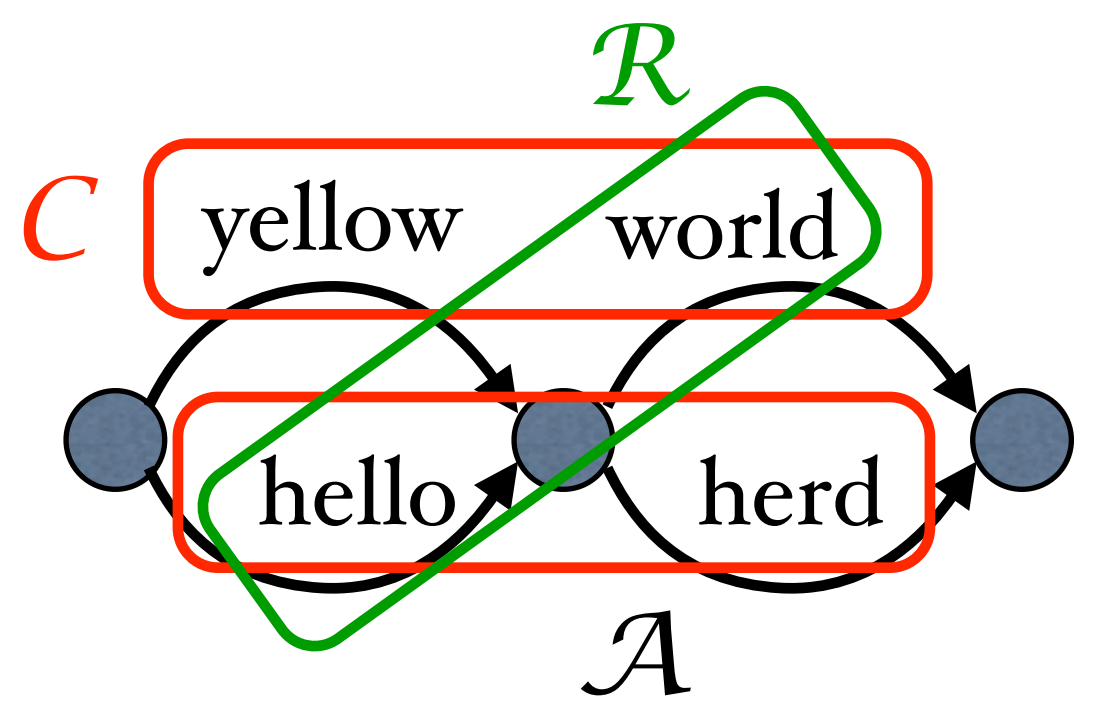
- Use lattices and Forward-Backward (FB) algorithm to break Equ. (1) into lattice arc components shared by many strings in set
- Use decoder to generate both reference lattices  $\mathcal{R}$  & recognition lattices  $\mathcal{A}$  for each utterance.

Note: In general,  $\mathcal{A}$  may contain  $\mathcal{R}$  as well as incorrect competitors  $\mathcal{C}$ .

- How to model competitor lattice  $\mathcal{C} = \mathcal{A} - \mathcal{R}$  that *excludes* the reference strings  $\mathcal{R}$  from the recognition lattice  $\mathcal{A}$ ? (need this for MCE)

→ Use Macherey et al. (2005) proposal:

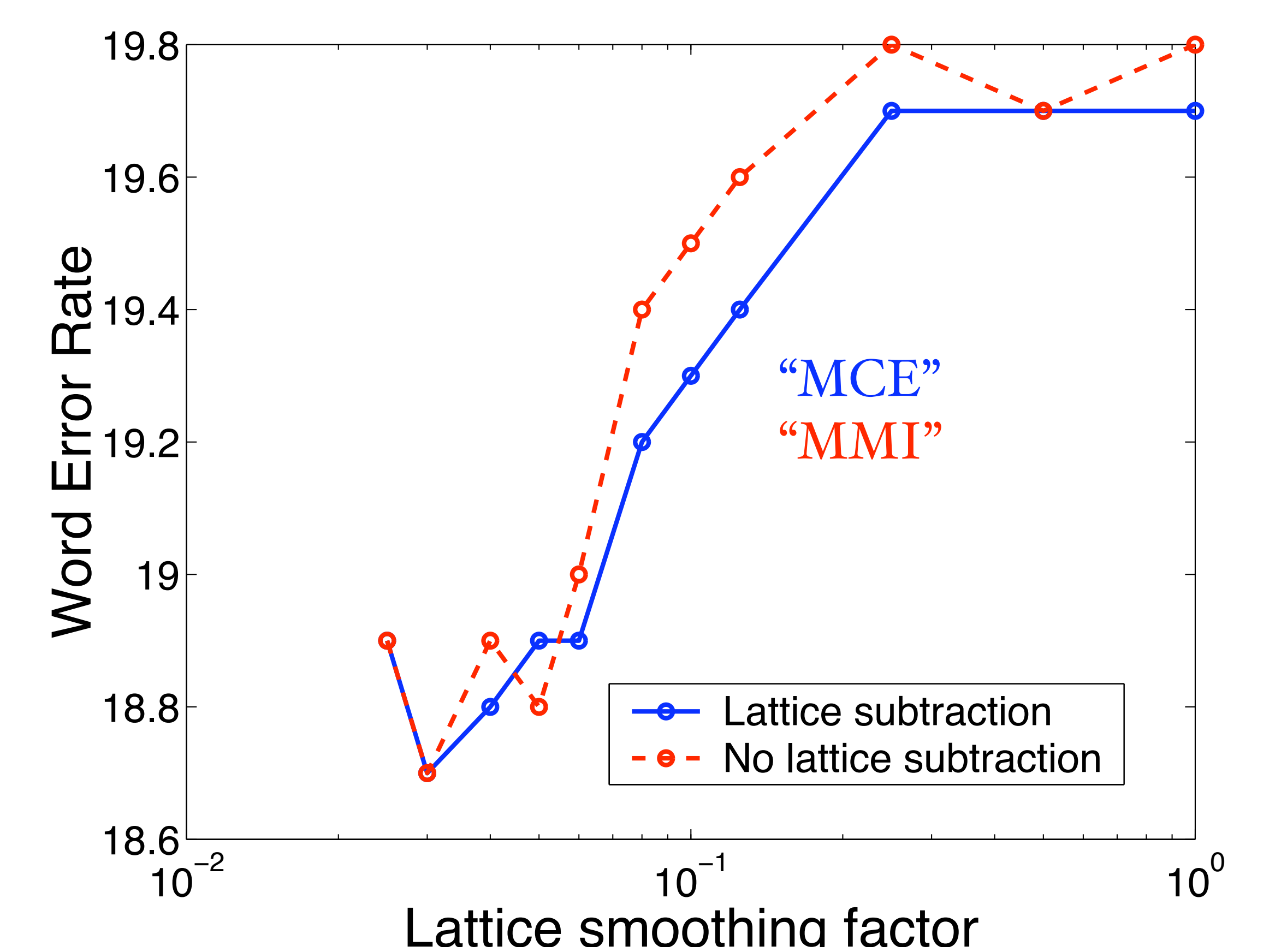
1. Identify  $\mathcal{R}$  within  $\mathcal{A}$  (WFST composition!)
2. FB on recognition lattice  $\mathcal{A}$ 
  - Calculate  $FB\_Score(arc)$  for all arcs in  $\mathcal{A}$ , and overall  $FB\_Score(\mathcal{A})$
3. FB on reference lattice  $\mathcal{R}$ 
  - Calculate  $FB\_Score(arc)$  for all arcs in  $\mathcal{R}$ , and overall  $FB\_Score(\mathcal{R})$
4. Set  $arc(occupancy) = FB\_Score(arc) / (FB\_Score(\mathcal{A}) - FB\_Score(\mathcal{R}))$  for all arcs in both  $\mathcal{A}$  and  $\mathcal{R}$
5. Use arc occupancies from part 4 to calculate all state-level derivatives/statistics (for all state means, covariances and mixing weights)
  1. *Add* all state-level derivatives for  $\mathcal{A}$  to accumulators
  2. *Subtract* all state-level derivatives for  $\mathcal{R}$  from accumulators.



- WERs, with (MCE) and without (MMI) numerical lattice subtraction:

| Gaussians        | ML   | MCE         | MMI         |
|------------------|------|-------------|-------------|
| 5000 x 32 (diag) | 21.6 | 18.7 (13.4) | 18.7 (13.4) |
| 2000 x 16 (full) | 19.8 | 18.5 (6.6)  | 18.6 (6.1)  |

- Results for different lattice smoothing factors  $\psi$  (Equ. 1):



## Summary

- Training with full LM improved upon previous work with unigram LM
- String-based training: full covariance models >> diagonal models
- Diagonal covariances: lattice-based training >> string-based training
- Lattice-based training: small gains for full covariances vs. diagonal models
- Small but consistent (?) gains for lattice subtraction based “exact MCE” compared to not using subtraction (“approximated MCE” or MMI)