Sphinx Benchmark Report

Long Qin
Language Technologies Institute
School of Computer Science
Carnegie Mellon University

Overview

- Evaluate general training and testing schemes
  - LDA-MLLT, VTLN, MMI, SAT, MLLR, CMLLR
- Use default setup and existing tools
  - SphinxTrain-0.8, Sphinx3
- Focus on WER, running time was not measured
  - Experiments were performed on different server machines, it's not easy to directly compare the xRT
- Test on different data
  - Easy task (WSJ) vs. broadcast news
  - English vs. Mandarin

Outline

- The baseline training scheme
- LDA-MLLT
- VTLN
- MMI
- SAT
- CMLLR
- MLLR
- Experiments
- Discussion

Baseline Training Scheme

13-MFCC with Delta and Delta-Delta

Triphone model
3-state HMM
GMM observation distribution

Monophone model
3-state HMM
1-Gaussian or GMM
observation distribution

Decision tree clustering with
auto-generated questions
A few thousand states

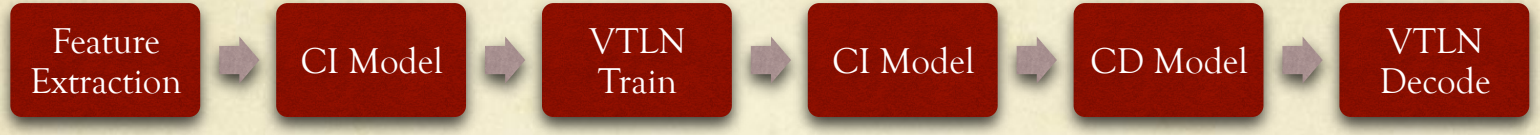
Force Alignment

- Force Alignment
  - Find the best alignment between speech and corresponding HMMs
- Goal
  - Possibly remove utterances with transcription errors or low quality recordings
  - Find appropriate pronunciations for words with multiple pronunciations
- Settings
  - $CFG_FORCEDALIGN = “yes”;
  - $CFG_FORCE_ALIGN_BEAM = 1e-60;
  - $CFG_FALIGN_CI_MGAU = “yes”/“no”;

LDA-MLLT

- LDA (linear discriminant analysis)
  - Find a linear transform of feature vector, so that class separation is maximized
  - Reduce feature dimension
- MLLT (maximum likelihood linear transform)
  - Minimize the loss of likelihood between full and diagonal variance model
  - Applied together with LDA
- In Sphinx
  - Each Gaussian is considered as one class
    - Easier to implement
    - Could also define state or phone as class
- Settings:
  - $CFG_LDA_MLLT = "yes";
  - $CFG_LDA_DIMENSION = 29;

VTLN



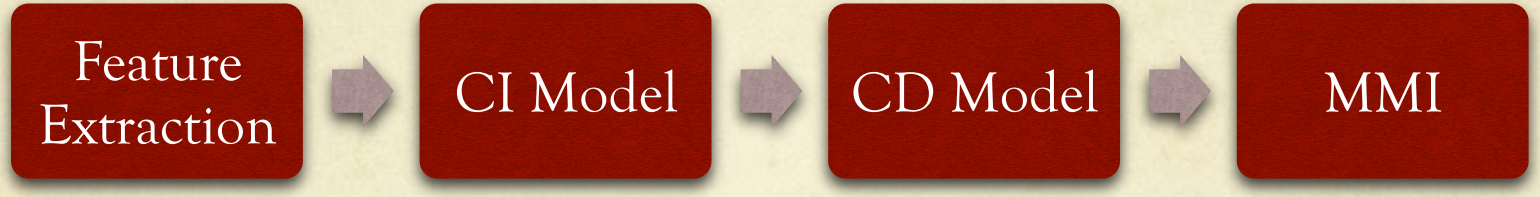
- VTLN (vocal tract length normalization)
  - Formant frequency is considered to have a linear relationship with the vocal tract length
  - Adjust vocal tract length for each speaker to an average length by warping their spectra
  - The warping factor:

$$\lambda = \arg\max P(O \mid X, \lambda_k)$$

- In Sphinx
  - Warping factor is estimated for each utterance using exhaust search
    - Could also estimate identical warping factor for each speaker
  - Warping factor should be estimated in both training and decoding

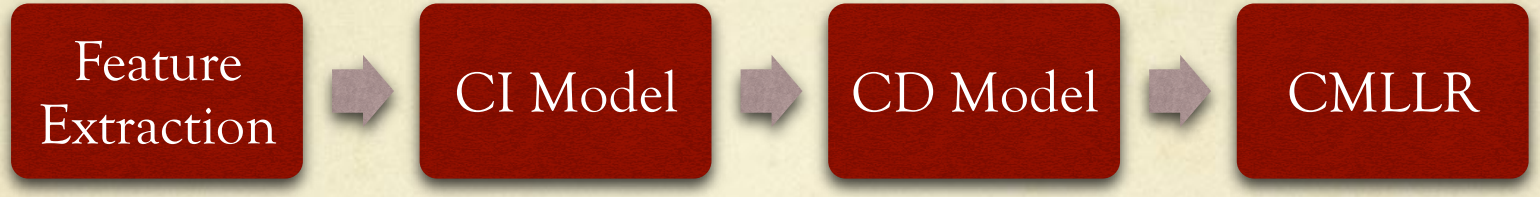
- Settings:
  - $CFG_VTLN = 'yes';
  - $CFG_VTLN_START = 0.70;
  - $CFG_VTLN_END = 1.40;
  - $CFG_VTLN_STEP = 0.05;

MMI



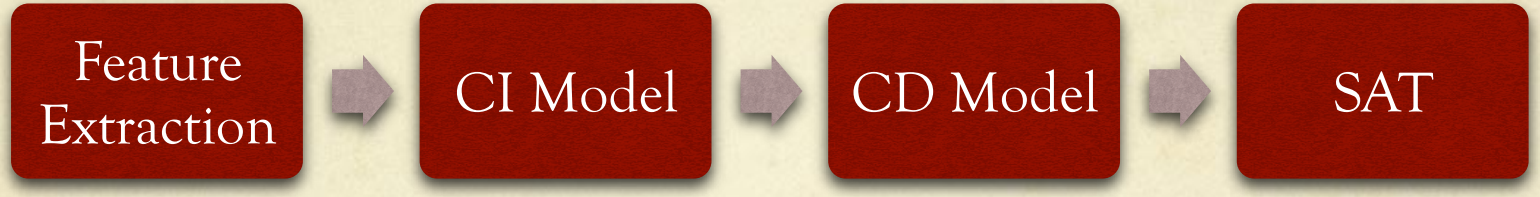
- MMI (maximum mutual information)
  - A discriminative training algorithm
  - Maximize the posterior probability of the true hypothesis
  - Training is time consuming
- Settings:
  - $CFG_MMIE_MAX_ITERATIONS = 4;
  - $CFG_MMIE_CONSTE = "3.0";
  - $CFG_LANGUAGEWEIGHT = "11.5";
    - The same as the language weight used in decoding
  - $CFG_LANGUAGEMODEL = "LMFILE";
    - A unigram or bigram LM

CMLLR



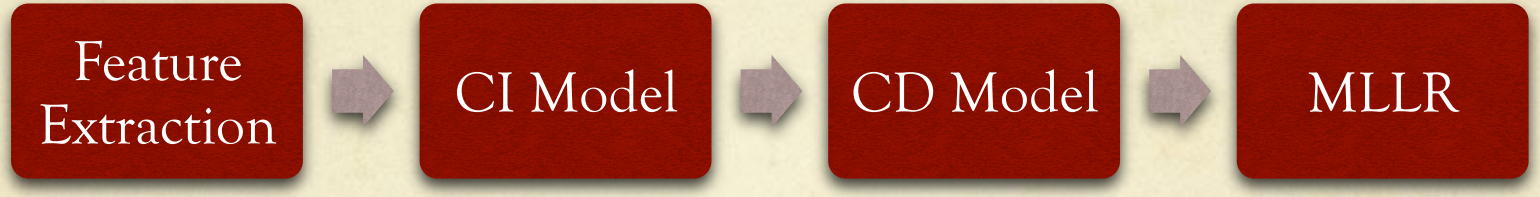
- CMLLR (constraint maximum likelihood linear regression)
  - A speaker adaptation algorithm to modify speaker independent system towards new speaker using limited data
  - Use the same transform for both mean and variance, therefore usually require less data then MLLR
  - Could be formulated as a linear transform of input features
- In Sphinx
  - Use a single global transform to adapt the input features for each speaker
  - When accumulate counts, run BW with "-fullvar yes", "-2passvar no" and "-cmllrdump yes"
- Settings:
  - $CFG_DEC_DICTIONARY = "DECODING_DICTIONRY";
  - $CFG_DEC_LM = "DECODING_LANGUAGE_MODEL";

SAT



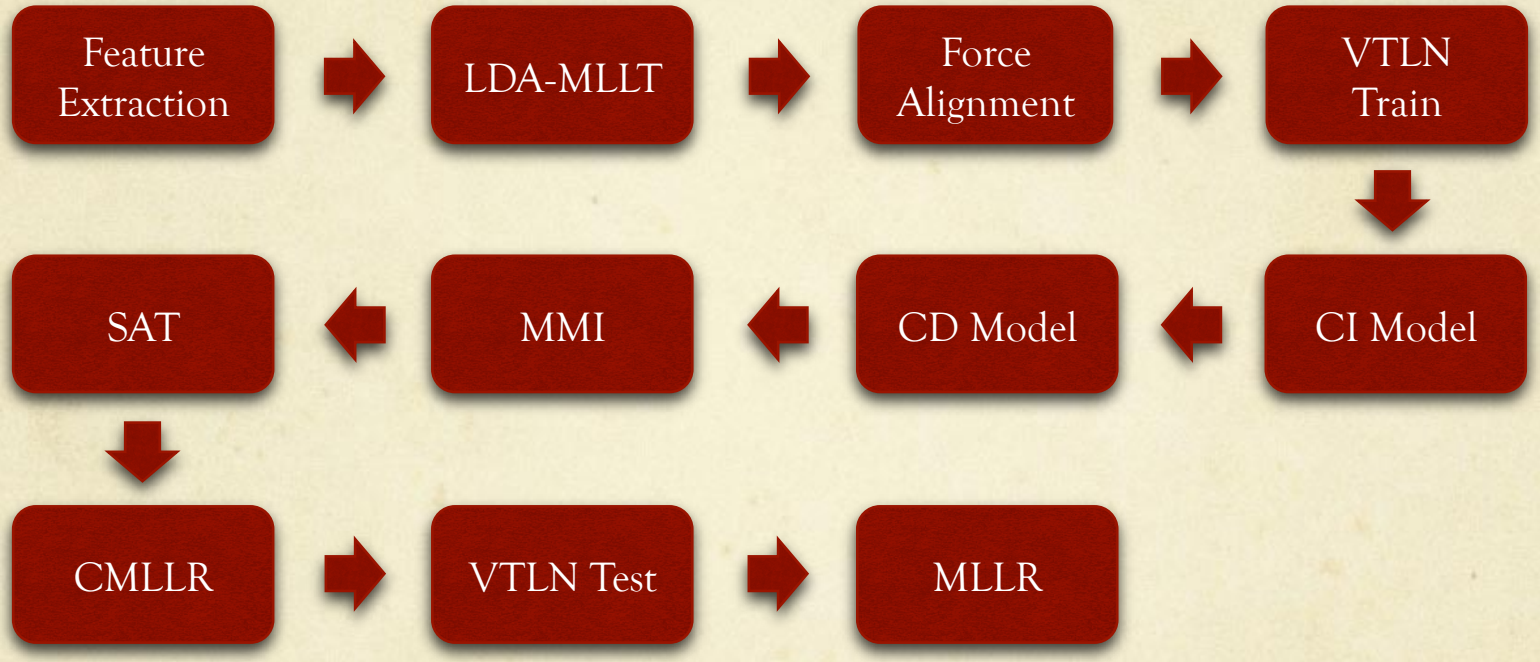
- SAT (speaker adaptive training)
  - Train a better speaker independent system
  - Apply CMLLR transforms to training features
  - Re-estimate the CMLLR transforms every iteration
- In Sphinx
  - SAT is applied after training a fairly good ML/MMI model
  - Need to split the training control and reference files into smaller files for each speaker (make_speaker_lists.py)
- Settings:
  - $CFG_SAT_DIR = "$CFG_BASE_DIR/sat";

MLLR



- MLLR (maximum likelihood linear regression)
  - Another speaker adaptation algorithm
  - Adjust mean and/or covariance to maximize the likelihood of the adaptation data
- In Sphinx
  - Adapt mean in default
    - Could also adapt covariance
  - Use a single global transform for all models
    - Could have multiple transforms for different classes of models
- Settings
  - Applied during decoding
    - Get hypotheses of the testing data from the first pass decoding
    - Using those hypotheses and testing data to estimate transforms and update model parameters
      - During bw run, must set "-2passvar no"
    - Decode again using the adapted model
  - It's the same procedure when we apply CMLLR/VTLN in decoding

Overall System Framework



Data

Training		Testing	LM
WSJ0	15-hour	Nov. 92 5k and 20k Dev/Eval	standard Trigram
WSJ0+1	82-hour		
BN	138-hour	HUB4-96 Dev/Eval (with data in all different environments)	Trigram from BN 92-97 LM data
Mandarin BN	128-hour	RT04-Eval	Trigram from Chinese Gigaword

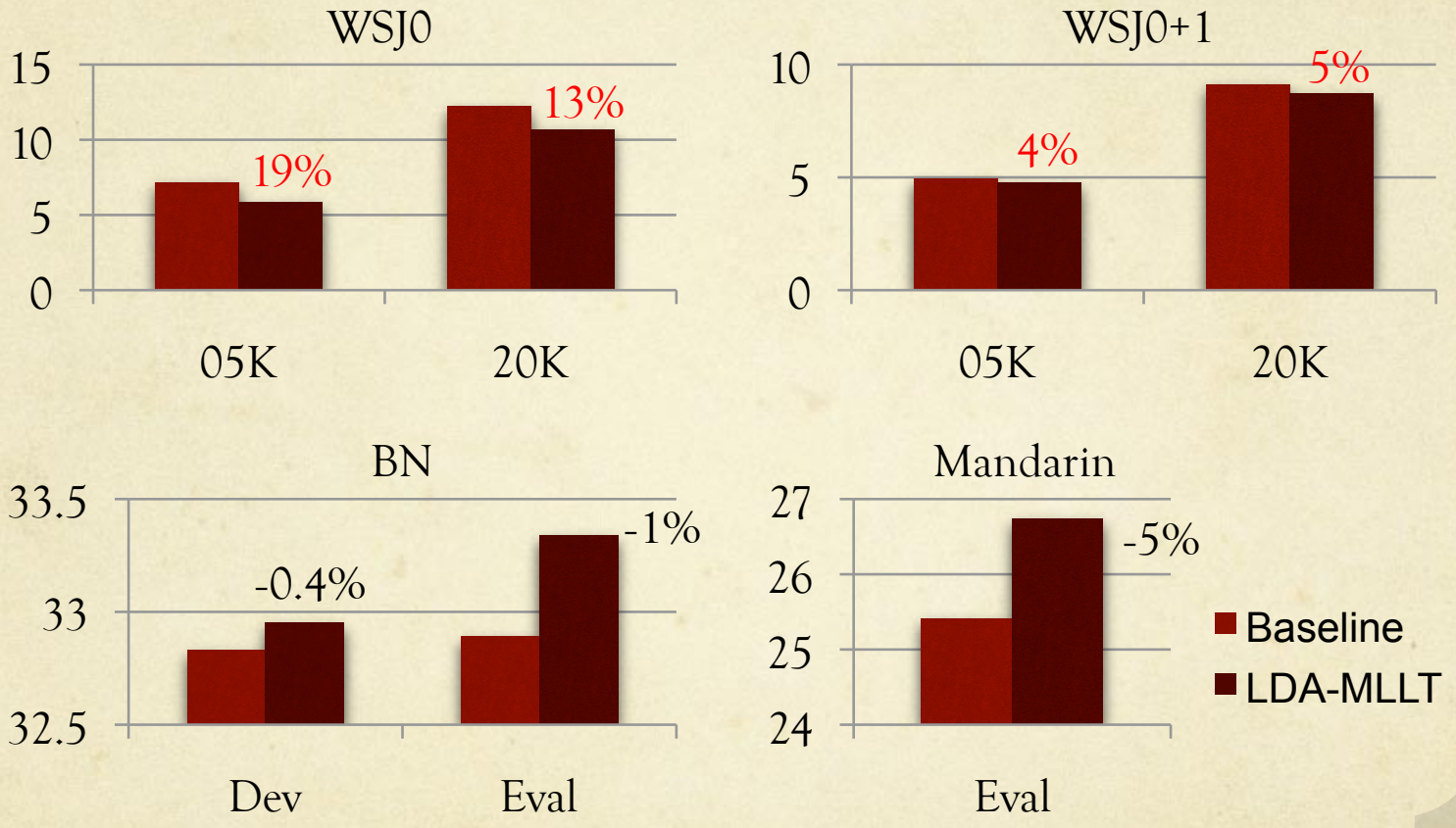
Baseline Settings

- Force Alignment
  - Could use multiple-Gaussian CI model
    - A little bit better, more computation
- Linguistic Questions
  - If available
  - Or use auto-generated questions
- Decoding
  - lw=11.5, beam=1e-100, wbeam=1e-80, wip=0.2
- Mixtures and States
  - WSJ0: 16 mixtures, 2000 tied-states
  - WSJ0+1: 32 mixtures, 4000 tied-states
  - BN: 32 mixtures, 5000 tied-states
  - Mandarin: 32 mixtures, 4000 tied-states

Baseline Results

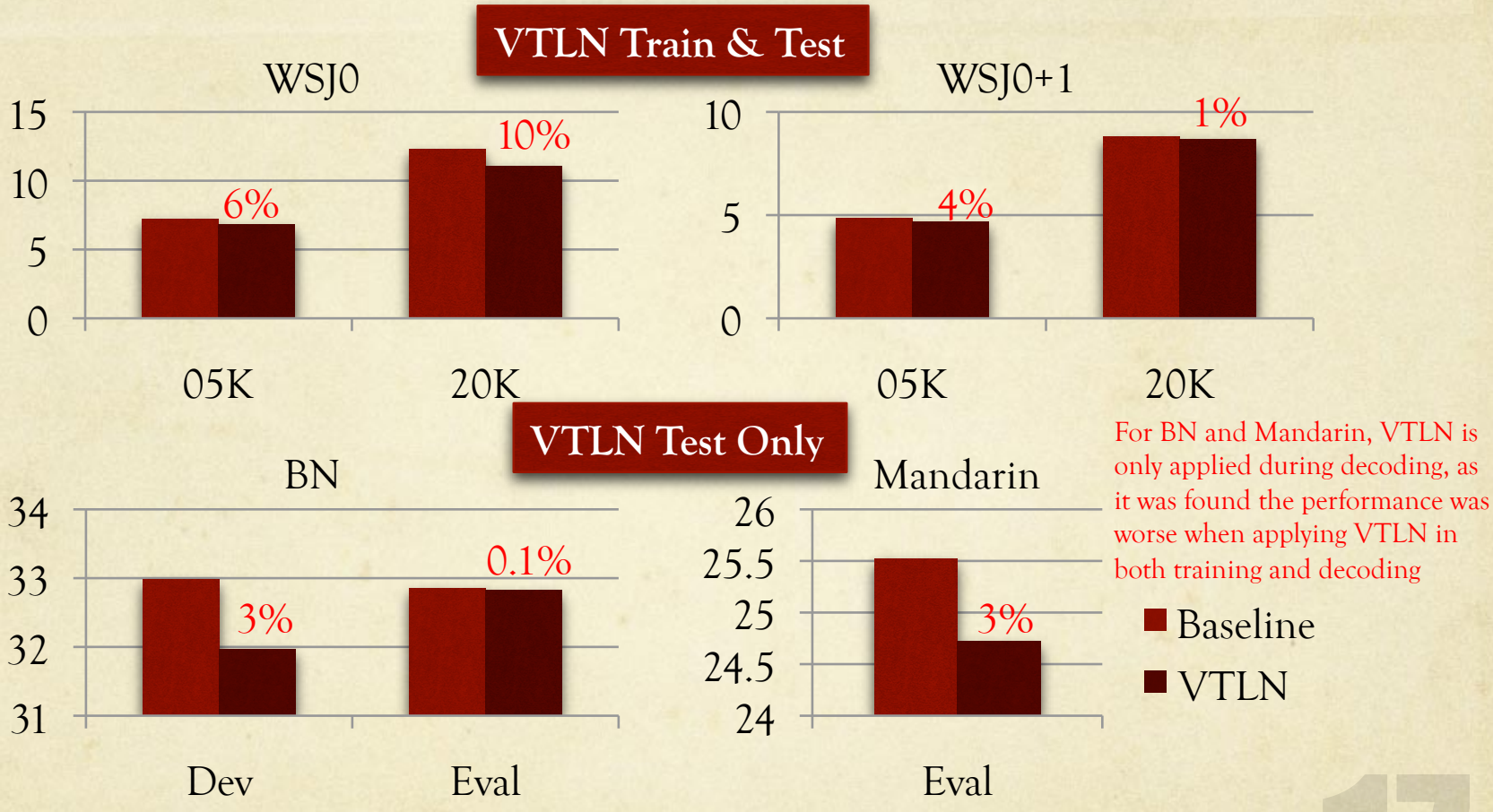
Data	WER (%)	
	Dev	Eval
WSJ0	7.62 (5k) 12.84 (20k)	6.85 (5k) 11.69 (20k)
WSJ0+1	5.50 (5k) 9.80 (20k)	4.18 (5k) 7.78 (20k)
BN	32.98	32.85
Mandarin	-----	25.35

LDA-MLLT Results



Comment: may work better on simple tasks with high quality data, but others (Joao Miranda) had tried it on noisy data, which also helped a lot. It works on telephone conversation tasks too.

VTLN Results

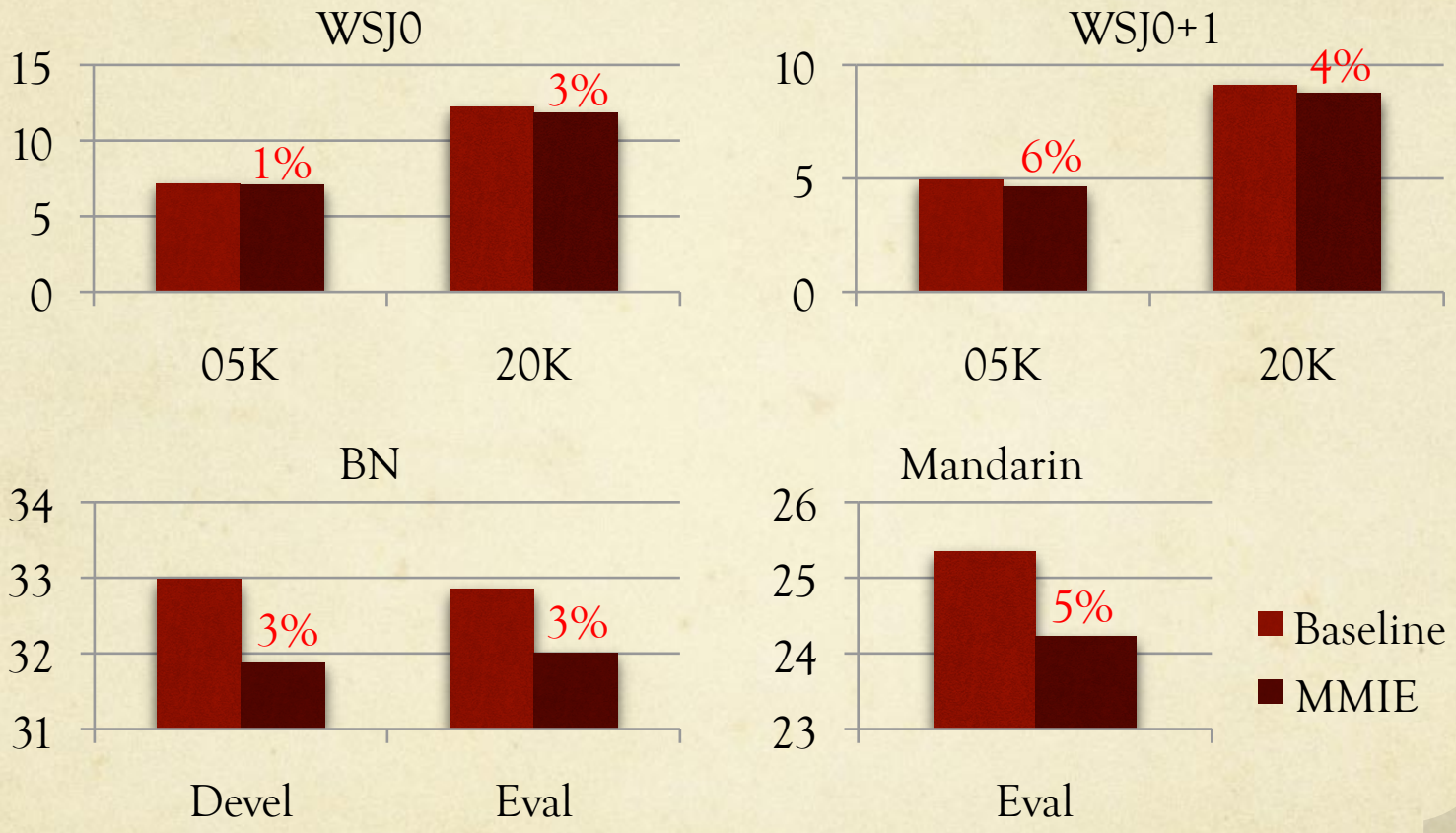


For BN and Mandarin, VTLN is only applied during decoding, as it was found the performance was worse when applying VTLN in both training and decoding

To be noticed:

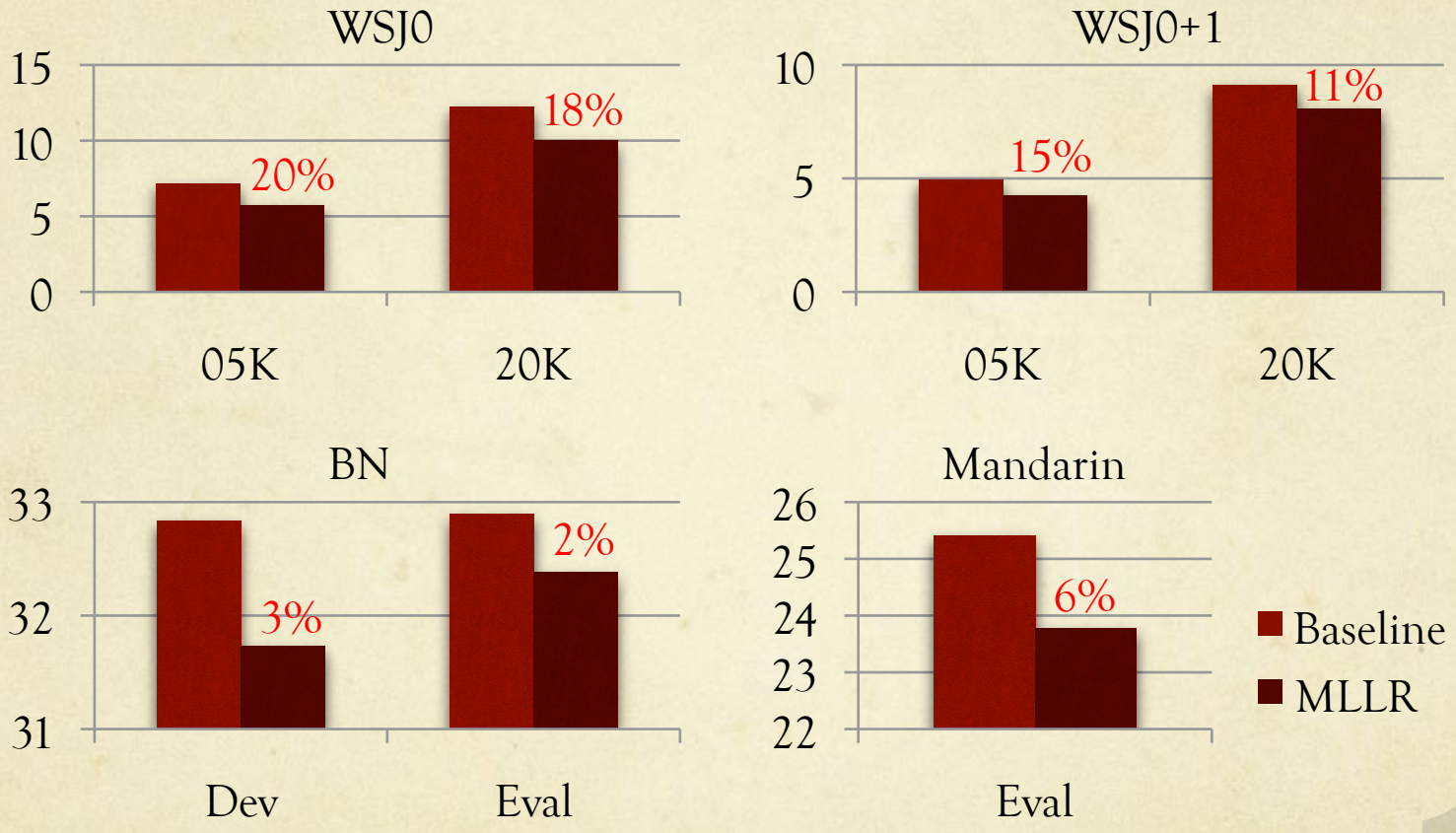
- the red numbers in the graph is the relative improvement over the baseline
- to have a graph without too many bars, the WSJ 5K/20K results are the average of the the Dev and Eval results

MMI Results



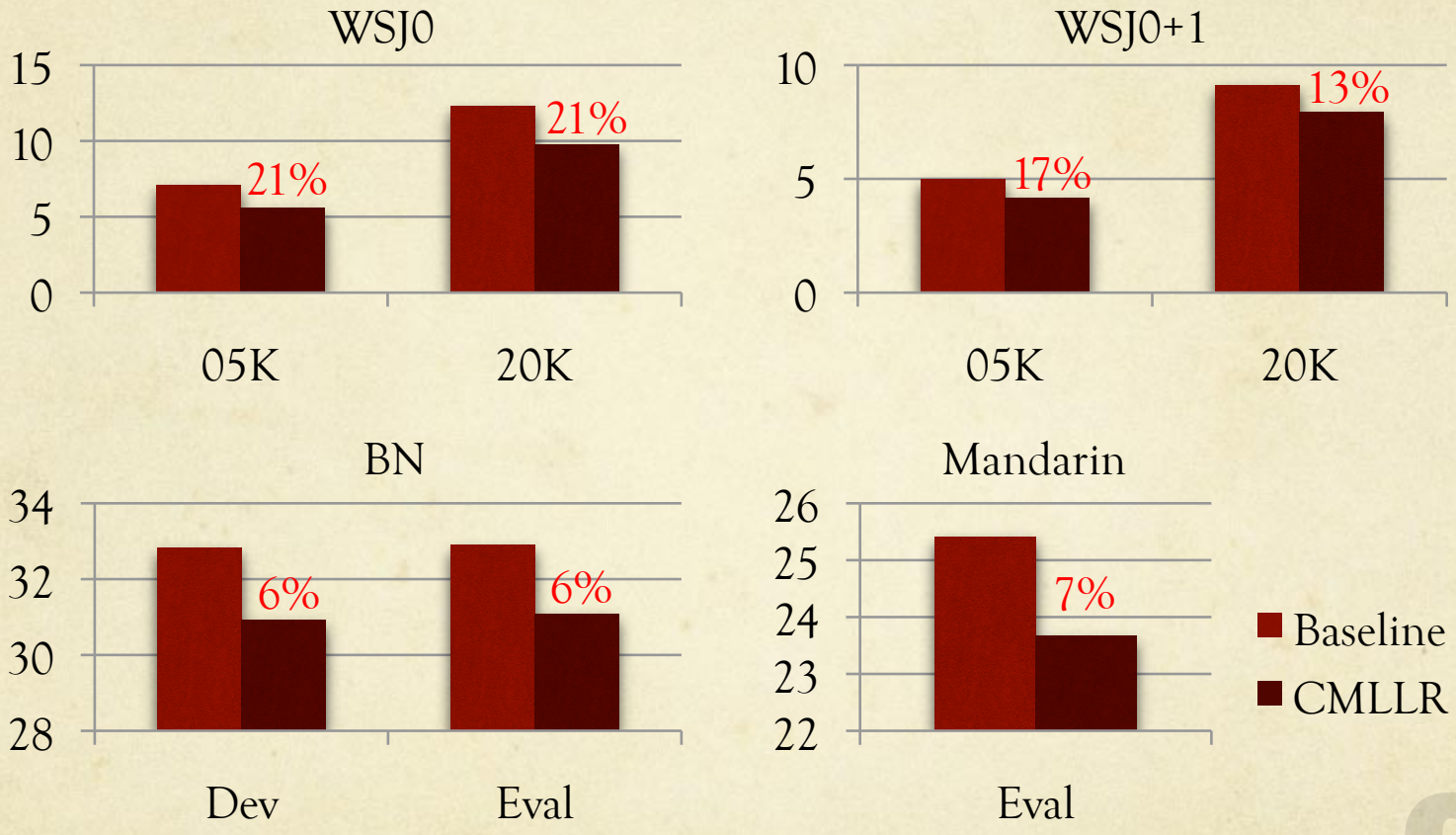
Comment: the results are not as good as I got from the lattice pruning experiments, where I used smaller lattices; try smaller beam widths when generate lattices, such as $beam = $wbeam = 1e-70, should be better and faster. Also try to use a bigram instead of unigram when generating lattices.

MLLR Results



Comment: works pretty good especially when the first path hypotheses are accurate; could use the second path hypotheses train a better transform and iteratively do it to get the best number

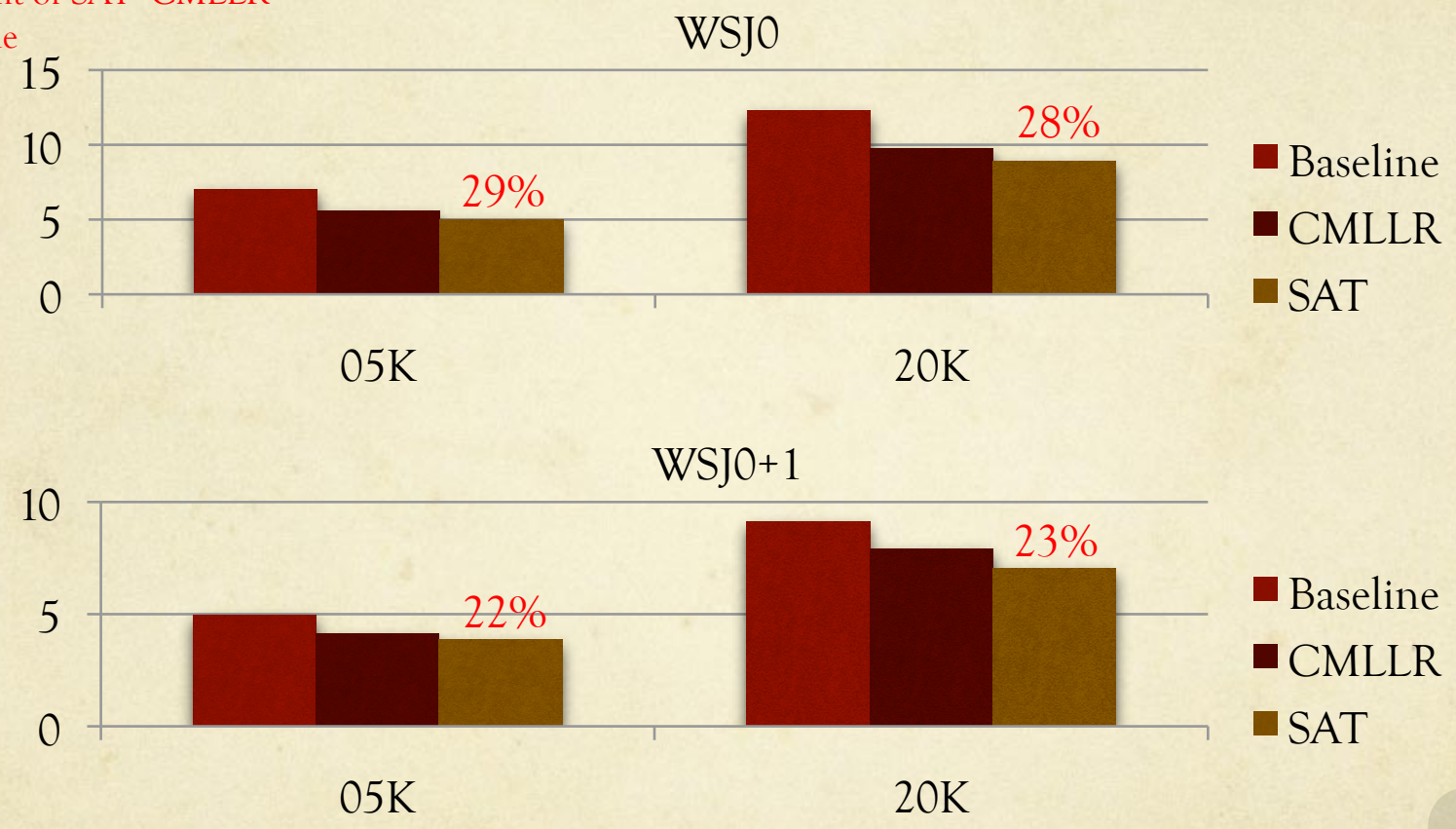
CMLLR Results



Comment: has similar performance as MLLR, slightly better in BN

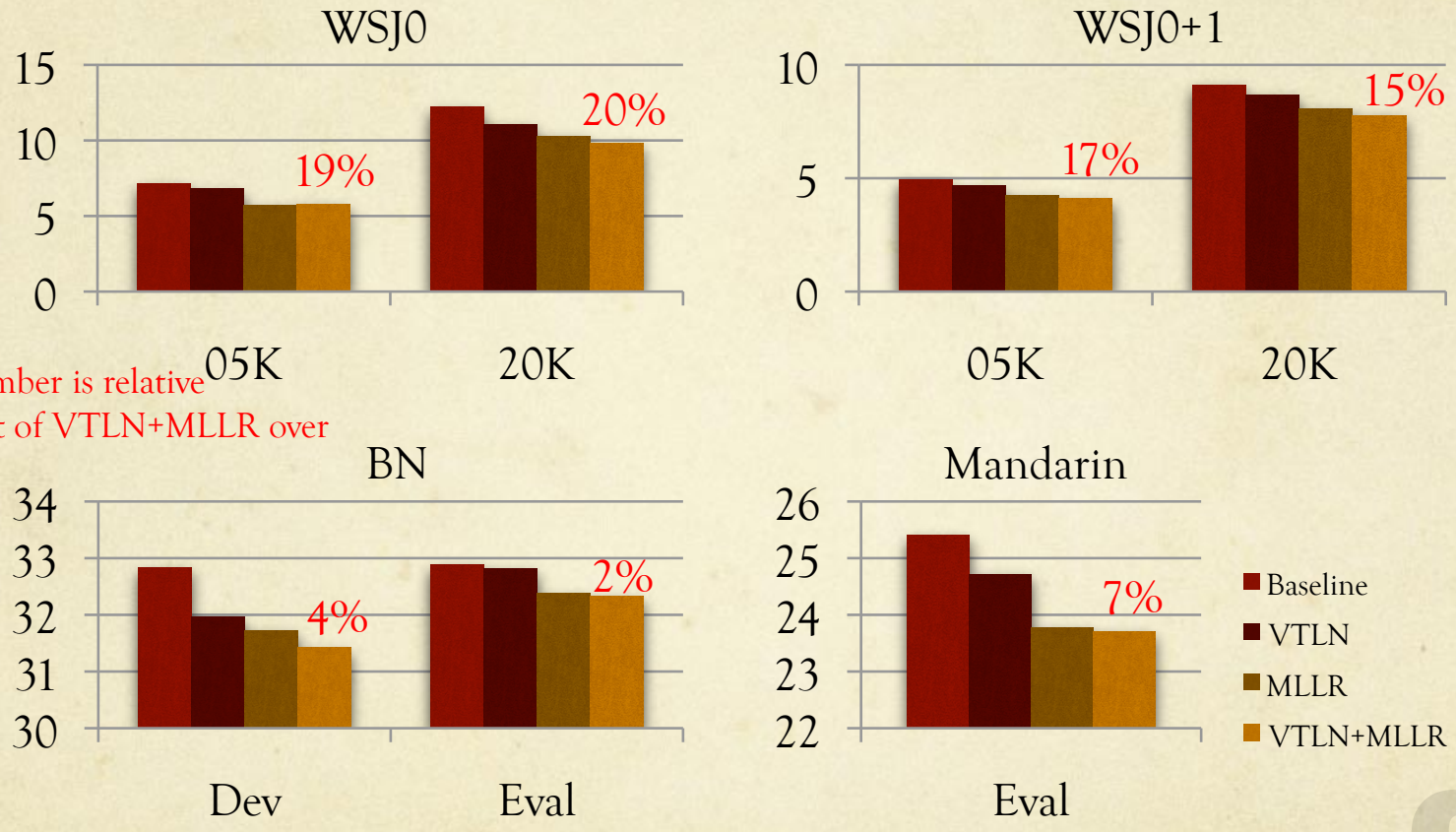
SAT Results

Here the number is relative improvement of SAT+CMLLR over baseline



Comment: SAT + CMLLR decoding is very effective, which usually gives 10% improvement over CMLLR decoding only. When estimating CMLLR transform, it's better to start from a very good hypothesis such as the CMLLR+MLLR decoding result.

VTLN + MLLR Results

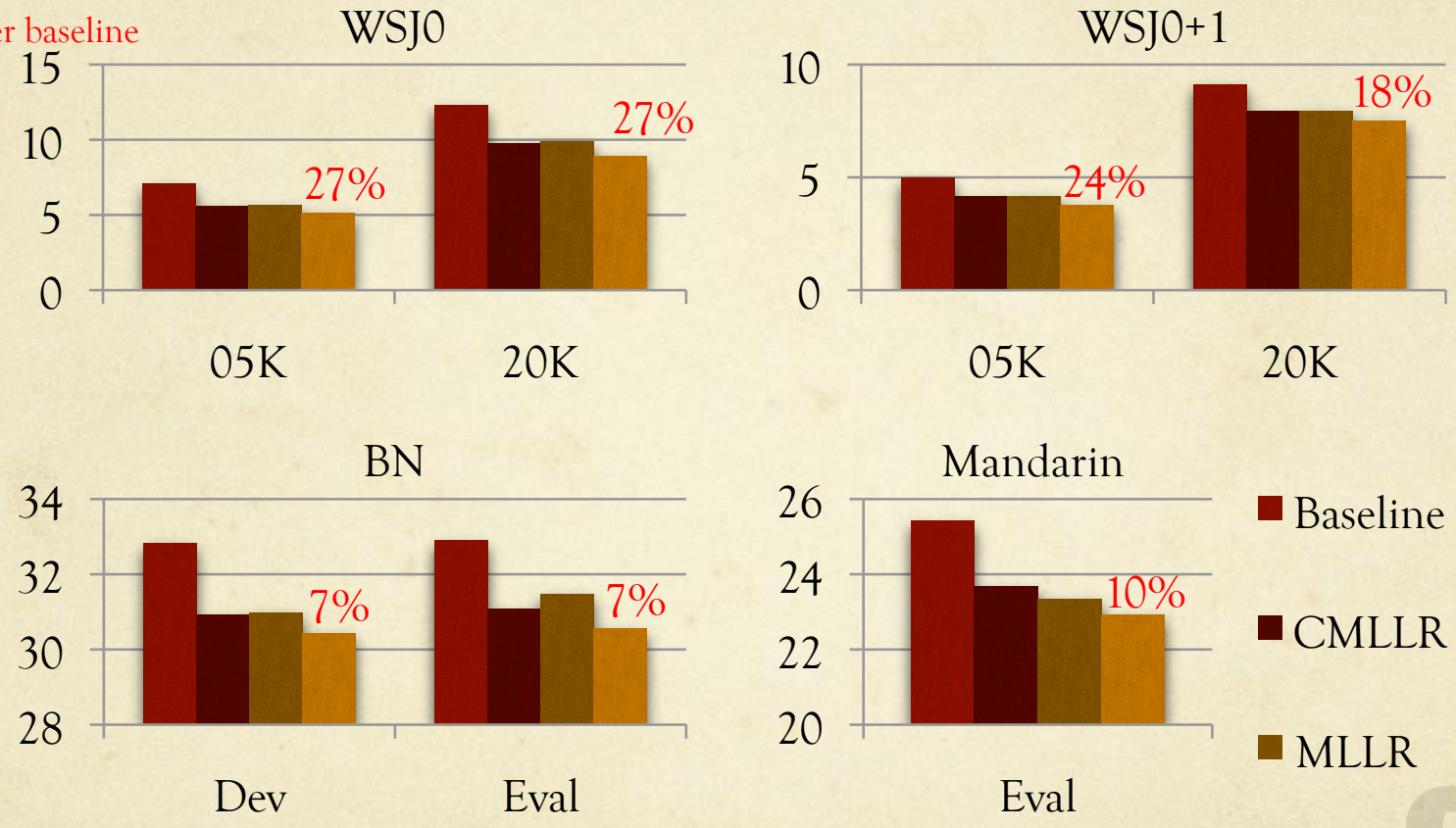


Here the number is relative improvement of VTLN+MLLR over baseline

Comment: the improvement is additive, but quite small compared to perform MLLR only

CMLLR+MLLR Results

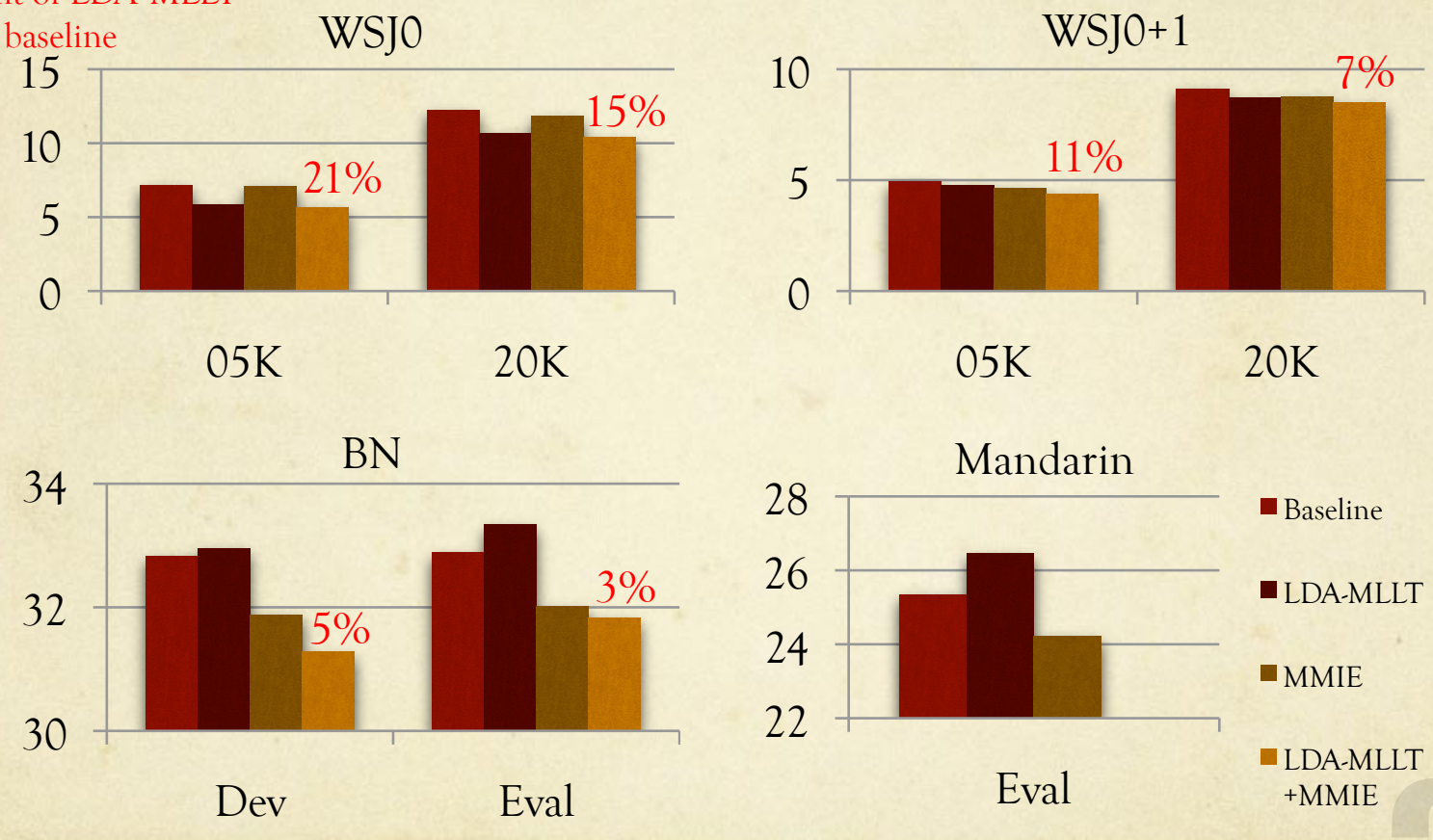
Here the number is relative improvement of CMLLR +MLLR over baseline



Comment: CMLLR+MLLR further improves the WER!

LDA-MLLT + MMI Result

Here the number is relative improvement of LDA-MLLT +MMI over baseline



Comment: MMIE gives solid improvement over LDA-MLLT (compare the 2nd bar and the 4th bar)

Summary

- LDA-MLLT
  - works pretty good on simple tasks with clear speech, not clear on hard tasks with noisy speech, needs more investigation
- VTLN
  - produces some improvement
- MMIE
  - produces ok/good improvement
  - requires large amount computation
- CMLLR
  - works pretty good, especially when first path hypotheses are very accurate
- MLLR
  - works similar to CMLLR
- SAT
  - produces solid improvement

Still Missing

- Better discriminative training technique
  - boosted-MMI
- Deep Neutral Network
  - Bottle Neck Feature (easier to adapt)
  - Hybrid Model (more improvement)