Single Pronunciation Dictionaries Construction and Performance

Thomas Hain

Machine Intelligence Lab
Cambridge University Engineering Department

September 5, 2003

Outline

- ➜ Motivation
- ➜ Pronunciation selection
  - ✘ Based on probabilities or frequencies
- ➜ Experiments
  - ✘ WSJ, CTS, BNE
- ➜ Explorations
  - ✘ Learning of pronunciation structure
  - ✘ Towards discriminative pronunciation selection
- ➜ Conclusions

Speech model construction

Speech (Sentence) models

➜ A series of probability density functions

Decision on PDFs based on

➜ Pronunciations in a dictionary

➜ Context

- ✘ phone and/or state level: using decision trees
- ✘ word-level: for example with multi-words ...

Question

➜ What information is needed to choose the appropriate PDF at the right time ?

Pronunciation representation

How much is achieved by a manual encoding of variation ?

Multi-modality (Substitutions)

➜ Combination
  ✘ Mixture models
➜ Divisive approach
  ✘ Decision trees

Durational variation (Insertions/Deletions)

➜ Forces multi-modality or broadening of distributions
  ✘ Mixture models
➜ Use phonemic context to decide on appropriate model handling deletion
  ✘ Decision trees
➜ ????

A step back: SPRONs

Given a dictionary with multiple pronunciations,
how to select the "best" ?

1. **Knowledge-based**

   ➜ Not discussed here

2. **Data-driven**

   ➜ Based on frequency of occurrence in alignment
   ➜ Distinction between words observed and those unseen

3. **Model-based**

   ➜ Best representation of acoustic subspace

Basic approach

Basic assumptions

- ➜ Simple substitutions of phonemes are irrelevant
- ➜ There exists a "canonical" phonemic representation of a word

Words observed in training data

- ➜ Merge substitution pairs
- ➜ Pick most frequent variant

Words not observed

- ➜ We need a criterion !

Given two phoneme sequences a and b,
which is the source $\mathbf{s}$ and which is target $\mathbf{t}$?

$$P(\mathbf{s}=a,\mathbf{t}=b) \lesseqgtr P(\mathbf{s}=b,\mathbf{t}=a)$$

Selection - Probabilistic

Simplify the criterion $P(\mathbf{s}=a,\mathbf{t}=b) \lessgtr P(\mathbf{s}=b,\mathbf{t}=a)$

1. Assume: **Equal priors** ($P(\mathbf{s}=a) = P(\mathbf{s}=b)$)

$$P(\mathbf{t}=b|\mathbf{s}=a) \lessgtr P(\mathbf{t}=a|\mathbf{s}=b)$$

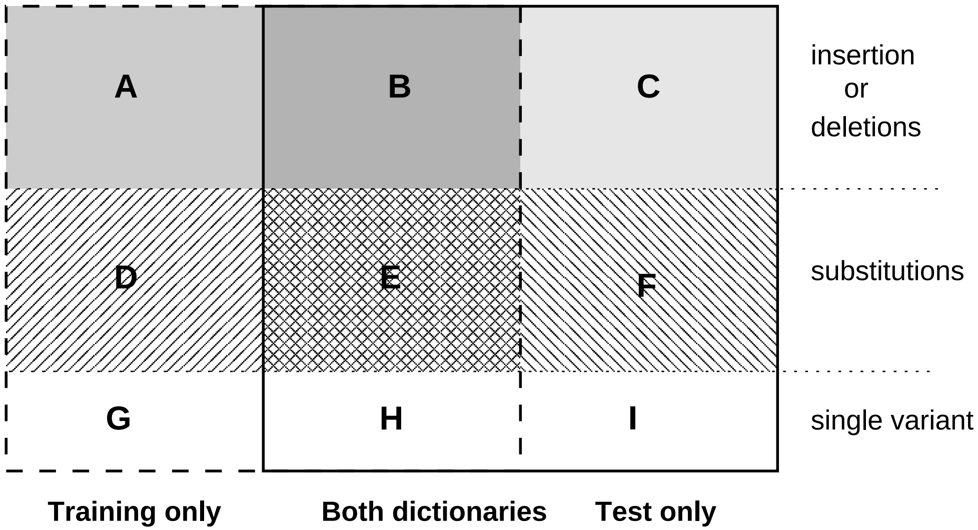
2. Assume: **Phone strings are DP-aligned**

k	aa	n	t	-	en	eh	n	t	el
k	aa	n	t	iy	n	eh	n	-	el

3. This allows to construct a simple model

$$P(\mathbf{t}|\mathbf{s}) = \prod_{i=1}^{M} P(t_i|\mathbf{t}_1^{t-1},\mathbf{s}) \approx \prod_{i=1}^{M} P(t_i|s_i)$$

Procedure



➜ Frequency based decision

1. Sets (D,E)
   decision + summing up counts
2. Sets (A,B)
   decision only

➜ Training of statistical model using sets (A,B,D,E)

$$P(t_i|s_i) = \frac{N(t_i, s_i)}{N(s_i)}$$

Need Add-One smoothing to avoid zero probabilities.

➜ Automatic decision using model for words in sets (C,F) using selection criterion

Selection - Frequency based

Further simplification of the selection:

$$P(\mathbf{t} = b | \mathbf{s} = a) \lessgtr P(\mathbf{t} = a | \mathbf{s} = b)$$

Take the counts as before

$$\prod_{i=1}^{M} \frac{N(b_i, a_i)}{N(a_i)} \lessgtr \prod_{i=1}^{M} \frac{N(a_i, b_i)}{N(b_i)} \qquad\qquad N(x, y) \neq N(y, x) \quad !$$

Taking the log

$$C_a + \sum_{i=1}^{M} \log N(a_i; b_i) \lessgtr C_b + \sum_{i=1}^{M} \log N(b_i; a_i)$$

and use $\log x \approx x - 1$

$$\sum_{i=1}^{M} N(a_i; b_i) \lessgtr \sum_{i=1}^{M} N(b_i; a_i)$$

Experiments - WSJ

- CU-HTK dictionary base
  - is the LIMSI'93 WSJ dictionary
  - Additions made using TTS system and checked manually
- WSJ setup
  - Straight-forward MLE system
  - 65k test dictionary: 1.11 pronunciations/word
  - 13k training dictionary: 1.18 pronunciations/word
- Dictionaries under investigation

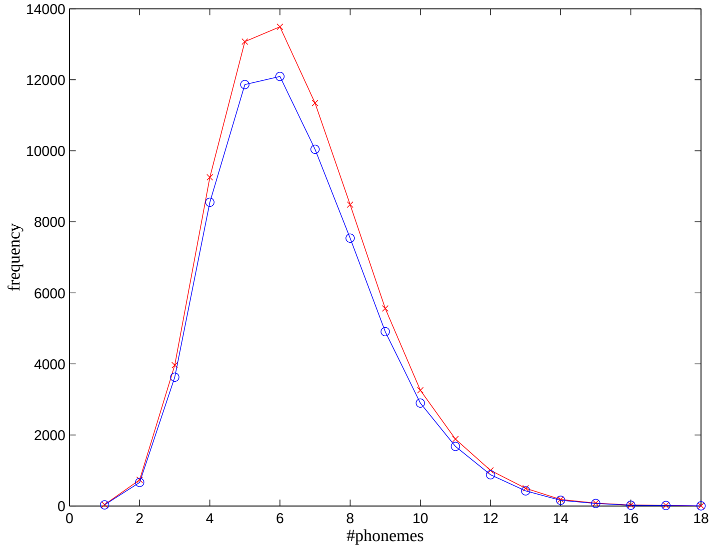
SPron1 Method P, using statistics from WSJ+Switchboard data
SPron2 Method P, using pronunciation statistics from WSJ only
SPron3 Purely random selection of pronunciations

SPRON selection - WSJ

➜ Models trained from scratch

Dict	#states	H1 Dev	H1 Eval	Average
MPron	6447	8.97	9.65	9.33
SPron1	6419	9.05	9.95	9.53
SPron2	6425	9.33	9.93	9.64
SPron3	6486	9.65	10.95	10.24

%WER results on the WSJ 1994 H1 Dev and eval test sets using different dictionaries for both training and test. #states denotes the number of clustered states in the model set.



Distribution of pronunciation lengths MPron/SPron1

Pronunciation variants in training and test

➜ Combining different strategies in training and test

- ✘ Using SPron1 dictionaries
- ✘ Only re-estimation(broken decision trees !)

Training Dict	Test Dict	H1 Dev	H1 Eval	Average
MPron	MPron	8.97	9.65	9.33
Mpron	SPron1	10.95	11.97	11.48
SPron1-ReEst	SPron1	9.37	10.31	9.86
SPron1-ReEst	MPron	9.07	9.50	9.30

%WERs on the WSJ H1 development and evaluation test sets. Results are obtained by rescoring trigram lattices.
All models are are state-clustered 12 mixture triphone models.

➜ SPron1-ReEst worse than re-clustering

➜ MPron information remains after re-estimation

Experiments - CTS

- Training sets
  - h5train03 (Swbd1 + Cell + CHE)
  - h5train03 + CTran data (Swbd2)

- Dictionaries
  - Training
    - *36k* (h5train03) 1.10 pronunciations/word
    - 40k (h5train03 + CTran) 1.10 pronunciations/word
  - Test
    - 54k (2002 dictionary) 1.10 pronunciations/word
    - 58k (2003 dictionary) 1.10 pronunciations/word

Comparing selection criteria

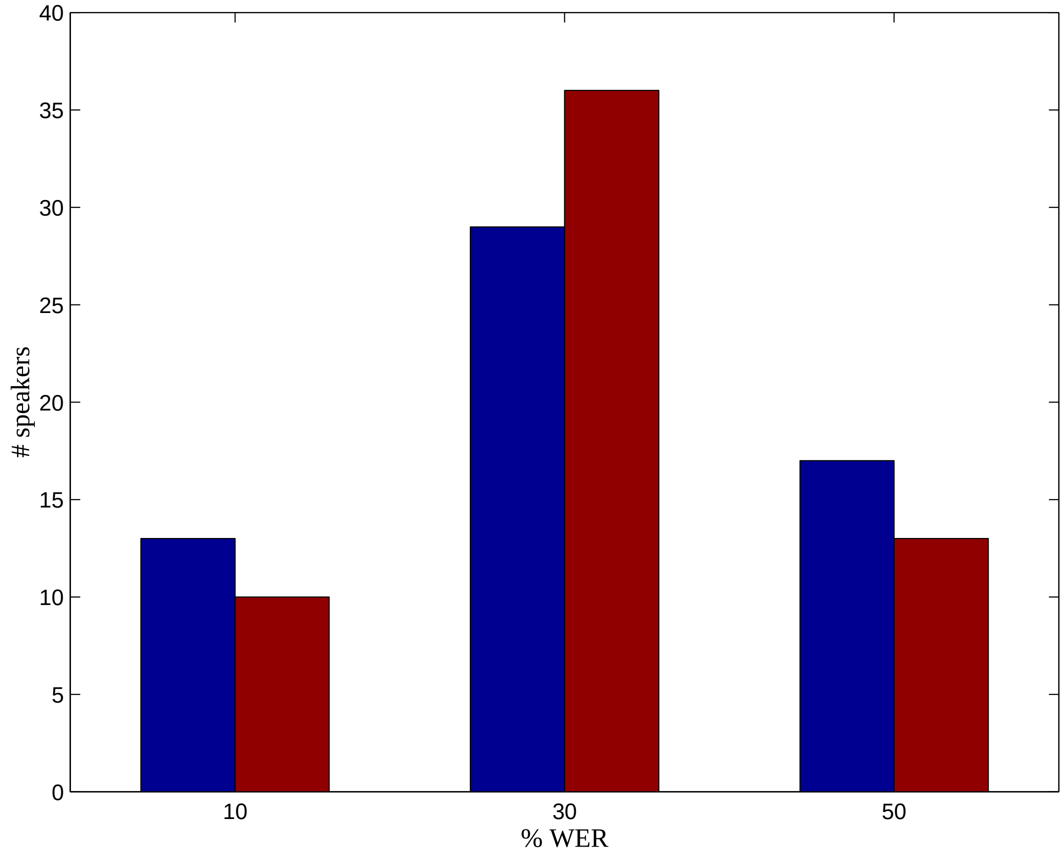
- ➜ Straight-forward MLE models trained on h5train03, 54k test dictionary
- ➜ Pronunciation statistics from BN training data + h5train03

Dict	SPRON Method	Swbd1	Swbd2	Cell	Average
MPron	-	26.4	41.2	40.7	36.0
SPron	F	25.8	39.6	39.2	34.8
SPron	P	25.5	40.2	39.4	34.9

%WERs obtained using decoding dev01 with a tg LM.
Models trained on the h5train03 training set (VTLN, 16 mixture components)

- ➜ Approximately similar performance on all test sets
- ➜ **Word level difference** MPron / SPron 21% (!) - mostly SF words

%WER difference per Speaker



Difference in word error rate per speaker on full dev01 set using PProbs
Red bars corresponds to results obtained with SPron+PProb
Blue bars with MPron+PProb.

Performance on eval03

➜ Adding pronunciation probabilities

- ✘ Based on frequency of variants, smoothing
- ✘ Pronunciation variants include silence thus probabilities for SPron dictionaries

➜ Performance of unadapted MLE/MPE systems (triphones/trigrams)

Setup	PronProb	MPron	SPron
MLE / 16mix		35.3	34.2
MLE / 16mix	×	34.4	33.8
28mix, HLDA, VarMix, MPE		27.4	26.9
28mix, HLDA, VarMix, MPE	×	27.2	26.8

➜ Regeneration of word lattices with SPron models brings 0.1%

Entropies - Effects of SProns

- Measuring the effect of reducing the number of pronunciations on uncertainty
  - Based on entropies $H(\mathbf{Q})$and $H\left(\mathbf{Q}|\mathbf{W}\right)$
- Using a prior distribution, either uniform or measured on data

	Perplexities 2^H			
Prior distribution	uniform		unigram	
Dictionary type	MPron	SPron	MPron	SPron
$H(\mathbf{W})$	54598	54598	2071.9	2071.9
$H(\mathbf{W}\|\mathbf{Q})$	1.128	1.125	1.082	1.065
$H(\mathbf{Q})$	85417.0	85369.2	3457.5	3201.2
$H(\mathbf{Q}\|\mathbf{W})$	1.765	1.758	1.834	1.672

- Effect of SProns only visible when using unigram prior

Experiments - BNE - dev03

- ➜ Similar setup to CTS experiments
  - ✘ Comparison unadapted MLE/MPE systems
  - ✘ Trained on ≈ 140 hours of data (bnetrain02)
  - ✘ Gender independent wide-band triphone models
  - ✘ Automatic segmentation (RT03 system)
  - ✘ Probabilistic SPron selection due to large number of test dictionary words not seen in training

MPron Dictionaries

- ➜ Training ($\approx 35k$ words)
  1.12 Prons/Word
- ➜ Test (59k words)
  1.10 Prons/Word

Setup	PProb	MPron	SPron
MLE		20.2	19.7
MLE	$\times$	19.0	18.9
HLDA, VarMix, MPE		15.3	14.8
HLDA, VarMix, MPE	$\times$	14.9	14.7

Where do we go from here ?

Observations

1. SPron dictionaries consistently yield similar or better performance on complex tasks with high acoustic confusability
2. Implicit modelling seems to allow better control on confusability
3. Suboptimal pronunciations for at least certain words

Probabilistic “pronunciation” networks

➜ Automatically learn variation

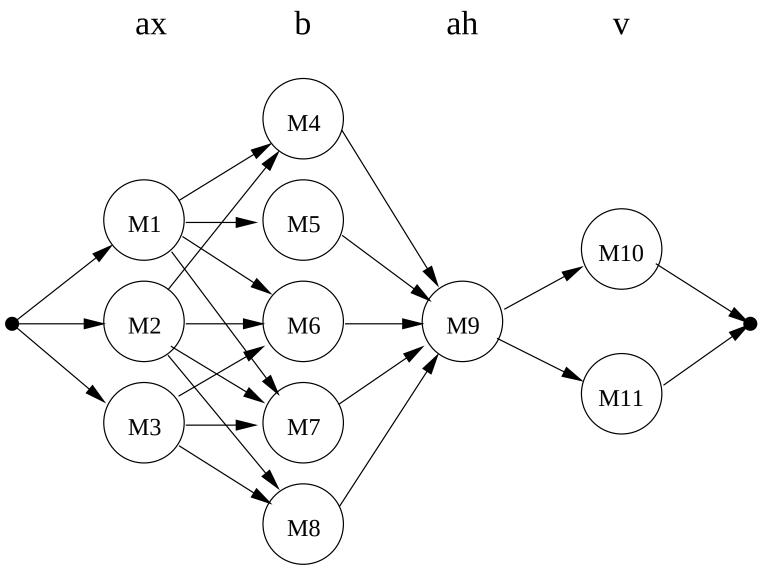
Discriminative pronunciation selection

➜ Find appropriate metrics for acoustic distance
➜ SPron generation as test case (non-discriminative)

Automatic learning of structure - HMS-HMMs

- Hidden model sequence models (HMS-HMMs)
  - One example for learning of structure
- Stochastic mapping between phoneme and HMM sequences
  - a "pronunciation model"
- Replaces phonetic decision trees
- integrated approach, training using EM framework
- allows modelling of temporal as well as substitution effects.

Network of models or states



SPron + HMS-HMMs - Performance on WSJ

- ➜ Same SPron dictionary (SPron1) as before
- ➜ HMS-HMM is initialised from the baseline HMM
  - ✘ same number of HMM parameters
  - ✘ modelling of substitutions only

		H1 Dev	H1 Eval	Average
HMM	Mpron	8.97	9.65	9.33
HMS-HMM	MPron	9.08	9.15	9.12
HMM	SPron1	9.05	9.95	9.53
HMS-HMM	SPron1	8.65	9.43	9.06

%WER results on the WSJ H1 Dev and eval test sets.

- ➜ Results on CTS indicate similar behaviour

Other criteria - Acoustic distance

When are pronunciations similar ?

- ➜ Pronunciation selection so far is based on ***symbolic similarity***
- ➜ ***Acoustic similarity*** is likely to be more appropriate

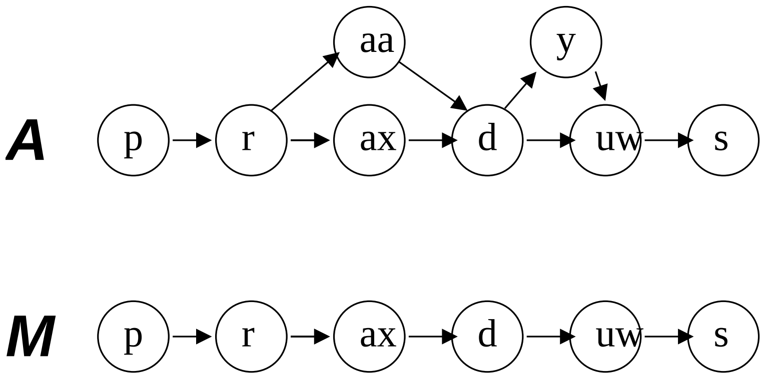
Pronunciation distance

- ➜ Acoustic similarity measurement based on a simulated data approach (Printz & Olsen 2002)
- ➜ HMM based
- ➜ computing $p(\mathcal{A}—\mathcal{M})$

Basic Idea

- ➜ Use model to represent the **acoustic word space**
- ➜ Pick the pronunciation with the minimal distance to that space

Pronunciation Selection



1. Form network with all prons of word w representing the acoustics $\mathcal{A}(w)$
2. Form network for pronunciation $q_i(w)$: $\mathcal{M}_i(w)$
3. **Expand** to triphone models, **context** from possible neighbouring phones and weight with phone bigram, **pruning**

Implementation

➜ Compute of $p(\mathcal{A}(w)|\mathcal{M}_i(w))$ using high-dimensional sparse matrix inversion

➜ Use posteriors (using pronunciation length normalisation and scaling)

$$P(q_i(w)|\mathcal{A}(w)) = \frac{p\left(\mathcal{A}(w)|\mathcal{M}_i(w)\right)^{\kappa} P(q_i(w))}{\sum_{l \in Q(w)} P(\mathcal{A}(w)|l)^{\kappa} P(l)}$$

➜ Pick pronunciation according to largest posterior

Results - MLE

Experiments on WSJ (same setup as before)

Dict	SPron Method	H1 Dev	H1 Eval	Average
MPron	-	8.97	9.65	9.33
SPron	P	9.05	9.95	9.53
SPron	Ac	9.18	9.99	9.60

Experiments on CTS (same as dev01 setup before)

Dict	SPRON Method	Swbd1	Swbd2	Cell	Average
MPron	-	26.4	41.2	40.7	36.0
SPron	F	25.8	39.6	39.2	34.8
SPron	Ac	25.6	40.0	39.5	35.0

➜ Similar performance to previous methods (note Swbd1 performance !)

➜ Preliminary results (pruning, scaling,...)

Conclusions

➜ Presented 3 methods for generating SPron dictionaries

✘ Probabilistic method gives best results so far

➜ SPron dictionaries give similar or better performance

✘ Better performance on more complex tasks
✘ Considerable improvement on MLE model sets
✘ Less so when comparing MPE models + PronProbs
✘ Automatic learning of pronunciation structure benefits

➜ SProns useful for system combination (considerable difference on word level)

➜ Future work

✘ Discriminative pronunciation selection