Discriminative Pronunciation Modeling

Joseph Keshet
Department of Computer Science
Bar-Ilan University

joint work with Hao Tang and Karen Livescu

Problem: Pronunciation variation

word probably

Problem: Pronunciation variation

word	probably
canonical pronunciation (baseform)	/pcl p r aa bcl b ax bcl b l iy/

Problem: Pronunciation variation

word	probably
canonical pronunciation (baseform)	/pcl p r aa bcl b ax bcl b l iy/
surface pronunciation (surface form)	[p r aa b iy] [p r aa l iy] [p r ay] [p ow ih] [p aa iy]

Previous Work

- Learn alternative pronunciations
  [Holter and Svendsen, 1999]
- Learn phonetic transformations
  [Riley et al., 1999, Hazen et al., 2005, Hutchinson and Droppo, 2011]
- Learn articulatory pronunciation models
  [Livescu and Glass, 2004, Jyothi et al., 2011]
- Learn alternative pronunciations with MCE
  [Vinyals et al., 2009, Korkmazskiy and Juang, 1997]

Contribution

- Propose a discriminative framework for pronunciation modeling
- Incorporate a large number of complex features
- Use large-margin learning

Lexical Access: Definition

$$[\text{p r aa l iy}] \mapsto ?$$

Lexical Access: Previous work

Experiments on a subset of Switchboard.

Model	Error Rate
lexicon lookup (from [Livescu, 2005])	59.3%

Lexical Access: Previous work

Experiments on a subset of Switchboard.

Model	Error Rate
lexicon lookup (from [Livescu, 2005])	59.3%
lexicon + Levenshtein distance	41.8%

Lexical Access: Previous work

Experiments on a subset of Switchboard.

Model	Error Rate
lexicon lookup (from [Livescu, 2005])	59.3%
lexicon + Levenshtein distance	41.8%
articulatory based DBN [Jyothi et al., 2011]	29.1%

Lexical Access: Previous work

Experiments on a subset of Switchboard.

Model	Error Rate
lexicon lookup (from [Livescu, 2005])	59.3%
lexicon + Levenshtein distance	41.8%
articulatory based DBN [Jyothi et al., 2011]	29.1%
Our approach	**15.2%**

Lexical Access: Goal

$$\begin{array}{ccc} & f & \\ \text{[p r aa l iy]} & \mapsto & \text{probably} \\ \mathbf{p} \in \mathcal{P}^* & & w \in \mathcal{V} \end{array}$$

$\mathcal{P}$	set of sub-word units
$\mathcal{P}^*$	set of all sequences of sub-word units
$\mathcal{V}$	vocabulary
w	word
$\mathbf{p}$	sequence of sub-word units

Model

We model $f : \mathcal{P}^* \to \mathcal{V}$ as

$$w^* = f(\mathbf{p}) = \operatorname*{argmax}_{w \in \mathcal{V}} \boldsymbol{\theta}^\top \phi(\mathbf{p}, w),$$

where $\boldsymbol{\theta} \in \mathbb{R}^n$ and $\phi(\mathbf{p}, w) : \mathcal{P}^* \times \mathcal{V} \to \mathbb{R}^n$.

For example, one of $\phi(\mathbf{p}, w)$ can be the Levenshtein distance between $\mathbf{p}$ and the canonical pronunciation of w.

Problem

Model

Features

- Dictionary Feature Function
- Length Feature Functions
- TF-IDF Feature Functions
- Articulatory Feature Functions

Learning

- Passive-Aggressive (PA)
- Strucural Support Vector Machine (SVM)

Experiments

Dictionary Feature Function

Define the dictionary feature function as

$$\phi_{\mathsf{dict}}(\mathbf{p}, w) = \mathbb{1}_{\mathbf{p} \in \mathit{pron}(w)},$$

where $\mathit{pron}(w)$ is the set of baseforms of w in the dictionary.

Dictionary Feature Function

Given a pronunciation dictionary:

⋮	
privacy	pcl p r ay1 ay2 v ax s iy
private	pcl p r ay1 ay2 v ax tcl t
pro	pcl p r ow1 ow2
probably	pcl p r aa bcl b ax bcl b l iy
problem	pcl p r aa bcl b l ax m
⋮	

$$\phi_{\text{dict}}([\text{pcl p r aa bcl b ax bcl b l iy}], \text{probably}) = 1$$
$$\phi_{\text{dict}}([\text{pcl p r aa bcl b ax bcl b l iy}], \text{problem}) = 0$$

Length Feature Functions

Suppose we have

w	probably										
$\mathbf{p}$	pcl	p	r	aa	bcl	b	l	iy			
pron(w)	pcl	p	r	aa	bcl	b	ax	bcl	b	l	iy

We want to see how the length of the surface form deviates from the baseform. In this case

$$\Delta\ell = -3.$$

Length Feature Functions

The length feature function is defined as

$$\phi_{\Delta\ell=r}(\mathbf{p}, w) = \mathbb{1}_{\Delta\ell=r} \otimes \mathbf{e}_w,$$

where $\Delta\ell = |\mathbf{p}| - |\mathbf{v}|$ for some $\mathbf{v} \in \mathit{pron}(w)$ and

$$\mathbf{e}_{w_i} = \begin{array}{l} w_1 \\ \vdots \\ w_{i-1} \\ w_i \\ w_{i+1} \\ \vdots \\ w_{|\mathcal{V}|} \end{array} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

TF-IDF Feature Functions

If I tell you /ih ng/ occurs at least once in the surface form, can you guess the word?

TF-IDF Feature Functions

If I tell you /ih ng/ occurs at least once in the surface form, can you guess the word?

according, accounting, adding, . . . , wondering, working, writing

TF-IDF Feature Functions

If I tell you /ih ng/ occurs at least once in the surface form, can you guess the word?

according, accounting, adding, . . . , wondering, working, writing

What if /ih ng/ occurs twice?

TF-IDF Feature Functions

If I tell you /ih ng/ occurs at least once in the surface form, can you guess the word?

according, accounting, adding, . . . , wondering, working, writing

What if /ih ng/ occurs twice?

bringing? singing?

TF-IDF Feature Functions

The "term" (sub-word unit) frequency is defined as

$$\mathrm{TF}_{\mathbf{u}}(\mathbf{p}) = \frac{1}{|\mathbf{p}| - |\mathbf{u}| + 1} \sum_{i=1}^{|\mathbf{p}|-|\mathbf{u}|+1} \mathbb{1}_{\mathbf{u}=\mathbf{p}_{i:i+|\mathbf{u}|-1}}.$$

Suppose $\mathbf{p} = [\text{p r aa l iy}]$. Then $\mathrm{TF}_{/\text{l iy}/}(\mathbf{p}) = \frac{1}{4}$.

Intuitively, if a sub-word unit has a high TF, then it is more discriminative.

TF-IDF Feature Functions

If I tell you /ih ng/ occurs at least once in the surface form, can you guess the word?

TF-IDF Feature Functions

If I tell you /ih ng/ occurs at least once in the surface form, can you guess the word?

according, accounting, adding, . . . , wondering, working, writing

TF-IDF Feature Functions

If I tell you /ih ng/ occurs at least once in the surface form, can you guess the word?

according, accounting, adding, . . . , wondering, working, writing

What if /z uw/ occurs?

TF-IDF Feature Functions

If I tell you /ih ng/ occurs at least once in the surface form, can you guess the word?

according, accounting, adding, . . . , wondering, working, writing

What if /z uw/ occurs?

zoo? zoology?

TF-IDF Feature Functions

The inverse "document" (word) frequency is defined as

$$\mathrm{IDF}_{\mathbf{u}} = \log \frac{|\mathcal{V}|}{|\mathcal{V}_{\mathbf{u}}|},$$

where $\mathcal{V}_{\mathbf{u}} = \{w \in \mathcal{V} \mid (\mathbf{p}, w) \in S, \mathbf{u} \in \mathbf{p}\}$.

Intuitively, if a sub-word unit is found in a small, specific set of words, then it is more discriminative.

TF-IDF Feature Functions

The final TF-IDF feature function for sub-word unit $\mathbf{u}$ is defined as

$$\phi_{\mathbf{u}}(\mathbf{p}, w) = (\mathrm{TF}_{\mathbf{u}}(\mathbf{p}) \times \mathrm{IDF}_{\mathbf{u}}) \otimes \mathbf{e}_w.$$

This feature function is also used in [Zweig et al., 2010].

Phonetic Alignment Feature Functions

Alignment 1

–	p	r	aa	–	–	l	iy
pcl	p	r	aa	bcl	b	l	iy

Alignment 2

–	p	r	aa	–	–	–	–	–	l	iy
pcl	p	r	aa	bcl	b	ax	bcl	b	l	iy

Phonetic Alignment Feature Functions

Turn these

–	p	r	aa	–	–	l	iy
pcl	p	r	aa	bcl	b	l	iy

–	p	r	aa	–	–	–	–	–	l	iy
pcl	p	r	aa	bcl	b	ax	bcl	b	l	iy

into this

$$\begin{array}{lcl} - & \rightarrow & \text{pcl} \\ \text{p} & \rightarrow & \text{p} \\ \text{r} & \rightarrow & \text{r} \\ \text{aa} & \rightarrow & \text{aa} \\ - & \rightarrow & \text{bcl} \\ - & \rightarrow & \text{b} \\ - & \rightarrow & \text{ax} \end{array}$$

Phonetic Alignment Feature Functions

Turn these

–	p	r	aa	–	–	l	iy
pcl	p	r	aa	bcl	b	l	iy

–	p	r	aa	–	–	–	–	–	l	iy
pcl	p	r	aa	bcl	b	ax	bcl	b	l	iy

into this

– $\rightarrow$ pcl
p $\rightarrow$ p
r $\rightarrow$ r
aa $\rightarrow$ aa
– $\rightarrow$ bcl
– $\rightarrow$ b
– $\rightarrow$ ax

Phonetic Alignment Feature Functions

Turn these

–	p	r	aa	–	–	l	iy
pcl	p	r	aa	bcl	b	l	iy

–	p	r	aa	–	–	–	–	–	l	iy
pcl	p	r	aa	bcl	b	ax	bcl	b	l	iy

into this

$$\begin{array}{lcl} - & \rightarrow & \text{pcl} \\ \text{p} & \rightarrow & \text{p} \\ \text{r} & \rightarrow & \text{r} \\ \text{aa} & \rightarrow & \text{aa} \\ - & \rightarrow & \text{bcl} \\ - & \rightarrow & \text{b} \\ - & \rightarrow & \text{ax} \end{array}$$

Phonetic Alignment Feature Functions

Turn these

–	p	r	aa	–	–	l	iy
pcl	p	r	aa	bcl	b	l	iy

–	p	r	aa	–	–	–	–	–	l	iy
pcl	p	r	aa	bcl	b	ax	bcl	b	l	iy

into this

$$\begin{array}{lcl} - & \rightarrow & \text{pcl} \\ \text{p} & \rightarrow & \text{p} \\ \text{r} & \rightarrow & \text{r} \\ \text{aa} & \rightarrow & \text{aa} \\ - & \rightarrow & \text{bcl} \\ - & \rightarrow & \text{b} \\ - & \rightarrow & \text{ax} \end{array}$$

Articulatory Feature Functions: Alignment

surface	s	s	eh	eh_n	eh_n	n	t	s	s	s
voicing	-	-	+	+	+	+	-	-	-	-
	s	s	eh	n	n	n	s	s	s	s
nasality	-	-	-	+	+	+	-	-	-	-
	s	s	eh	n	n	n	s	s	s	s
tongue body	u	u	u	p	p	u	u	u	u	u
	s	s	eh	eh	eh	n	n	s	s	s
tongue tip	cr	cr	cr	m	m	cl	cl	cr	cr	cr
	s	s	eh	eh	eh	n	n	s	s	s

Articulatory Feature Functions: Alignment

- We define alignment feature functions on the articulatory level similar to the phonetic alignments.
- Alignment is done with articulatory based Dynamic Bayesian Network [Livescu and Glass, 2004].

$$\phi_{\text{artic-align}}(\mathbf{p}, w) = \begin{array}{l} \text{lip-loc-lab} \rightarrow \text{lip-loc-den} \\ \text{lip-open-clo} \rightarrow \text{lip-open-wide} \\ \text{tongue-tip-den} \rightarrow \text{tongue-tip-alv} \\ \text{vel-clo} \rightarrow \text{vel-open} \\ \vdots \end{array} \begin{pmatrix} 0.5 \\ 0.1 \\ 0.3 \\ 0.2 \\ \vdots \end{pmatrix}$$

Articulatory Feature Functions: Log-likelihood

We also include the log-likelihood of the alignment as a feature,

$$\phi_{LL}(\mathbf{p}, w) = \frac{\mathcal{L}(\mathbf{p}, w) - h}{k},$$

where

$\mathcal{L}(\mathbf{p}, w)$	log-likelihood
h	shift
k	scale

Articulatory Feature Functions: Asynchrony

sense /s eh n s/ $\rightarrow$ [s eh_n n t s]

surface	s	s	eh	eh_n	eh_n	n	t	s	s	s
voicing	-	-	+	+	+	+	-	-	-	-
	s	s	eh	n	n	n	s	s	s	s
nasality	-	-	-	+	+	+	-	-	-	-
	s	s	eh	n	n	n	s	s	s	s
tongue body	u	u	u	p	p	u	u	u	u	u
	s	s	eh	eh	eh	n	n	s	s	s
tongue tip	cr	cr	cr	m	m	cl	cl	cr	cr	cr
	s	s	eh	eh	eh	n	n	s	s	s
asynchrony				1	1		1			

Articulatory Feature Functions: Asynchrony

sense /s eh n s/ → [s eh_n n t s]

surface	s	s	eh	eh_n	eh_n	n	t	s	s	s
voicing	-	-	+	+	+	+	-	-	-	-
	s	s	eh	n	n	n	s	s	s	s
nasality	-	-	-	+	+	+	-	-	-	-
	s	s	eh	n	n	n	s	s	s	s
tongue body	u	u	u	p	p	u	u	u	u	u
	s	s	eh	eh	eh	n	n	s	s	s
tongue tip	cr	cr	cr	m	m	cl	cl	cr	cr	cr
	s	s	eh	eh	eh	n	n	s	s	s
asynchrony				1	1		1			

Articulatory Feature Functions: Asynchrony

sense /s eh n s/ $\rightarrow$ [s eh_n n t s]

surface	s	s	eh	eh_n	eh_n	n	t	s	s	s
voicing	-	-	+	+	+	+	-	-	-	-
	s	s	eh	n	n	n	s	s	s	s
nasality	-	-	-	+	+	+	-	-	-	-
	s	s	eh	n	n	n	s	s	s	s
tongue body	u	u	u	p	p	u	u	u	u	u
	s	s	eh	eh	eh	n	n	s	s	s
tongue tip	cr	cr	cr	m	m	cl	cl	cr	cr	cr
	s	s	eh	eh	eh	n	n	s	s	s
asynchrony				1	1		1			

Articulatory Feature Functions: Asynchrony

Define the asynchrony among articulatory variables feature functions as

$$\phi_{a \leq \mathrm{async}(\mathcal{F}_1, \mathcal{F}_2) < b}(\mathbf{p}, w) = \mathbb{1}_{a \leq \mathrm{async}(\mathcal{F}_1, \mathcal{F}_2) < b},$$

where

$\mathcal{F}_1$ and $\mathcal{F}_2$	sets of articulatory variables
$\mathrm{async}(\mathcal{F}_1, \mathcal{F}_2)$	the asynchrony between $\mathcal{F}_1$ and $\mathcal{F}_2$

Features: Big picture

$$\phi(\mathbf{p}, w) = \left[\begin{array}{c} \mathbb{1}_{\mathbf{p} \in pron(w)} \\ \hline \mathbb{1}_{a \leq \Delta\ell < b} \otimes \mathbf{e}_{\mathrm{a}} \\ \vdots \\ \mathbb{1}_{a \leq \Delta\ell < b} \otimes \mathbf{e}_{\mathrm{zero}} \\ \hline \mathrm{TF}_{\mathbf{u}}(\mathbf{p})\mathrm{IDF}_{\mathbf{u}} \otimes \mathbf{e}_{\mathrm{a}} \\ \vdots \\ \mathrm{TF}_{\mathbf{u}}(\mathbf{p})\mathrm{IDF}_{\mathbf{u}} \otimes \mathbf{e}_{\mathrm{zero}} \\ \hline - \to \mathrm{pcl} \\ \mathrm{p} \to \mathrm{p} \\ \mathrm{r} \to \mathrm{r} \\ - \to \mathrm{bcl} \\ \vdots \end{array}\right] \begin{array}{l} \\ \\ \left.\begin{array}{c} \\ \\ \\ \end{array}\right\} \#\text{ of ranges} \times |\mathcal{V}| \\ \\ \left.\begin{array}{c} \\ \\ \\ \end{array}\right\} \#\text{ of sub-word units} \times |\mathcal{V}| \\ \\ \left.\begin{array}{c} \\ \\ \\ \\ \\ \end{array}\right\} (|\mathcal{P}| + 1)^2 - 1 \end{array}$$

Features: Big picture

$$\phi(\mathbf{p}, w) = \left[\begin{array}{c} \left.\begin{array}{c} \text{lip-loc-lab} \to \text{lip-loc-den} \\ \text{lip-open-clo} \to \text{lip-open-wide} \\ \text{tongue-tip-den} \to \text{tongue-tip-alv} \\ \text{vel-clo} \to \text{vel-open} \\ \vdots \end{array}\right\} \sum_{i=1}^{7} |F_i|^2 \\ \hline \dfrac{\mathcal{L}(\mathbf{p}, w) - h}{k} \\ \hline \left.\begin{array}{c} \mathbb{1}_{a \le \text{async(tongue tip,tongue body)} < b} \\ \mathbb{1}_{a \le \text{async(lip,tongue)} < b} \\ \vdots \end{array}\right\} \begin{array}{c} \#\text{ of ranges } \times \\ \#\text{ of combinations} \end{array} \end{array}\right]$$

Learning: Passive-Aggressive (PA) [Crammer et al., 2006]

The goal is to find

$$\begin{aligned}\boldsymbol{\theta}_{t+1} = \operatorname*{argmin}_{\boldsymbol{\theta}} \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}_t\|_2^2 \\ \text{s.t. } \boldsymbol{\theta}^\top \phi(\mathbf{p}_i, w_t) - \boldsymbol{\theta}^\top \phi(\mathbf{p}_i, \hat{w}) \geq \mathbb{1}_{w_t \neq \hat{w}},\end{aligned}$$

where

$$\hat{w} = \operatorname*{argmax}_{w \in \mathcal{V}} \left[\mathbb{1}_{w_t \neq w} - \boldsymbol{\theta}^\top \phi(\mathbf{p}_t, w_t) + \boldsymbol{\theta}^\top \phi(\mathbf{p}_t, w) \right].$$

Learning: Structural Support Vector Machine (SVM)

Let $S = \{(\mathbf{p}_1, w_1), \ldots, (\mathbf{p}_m, w_m)\}$. The goal is find

$$\boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta}} \frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2 + \sum_{i=1}^{m} \ell(\boldsymbol{\theta}; \mathbf{p}_i, w_i),$$

where

$$\ell(\boldsymbol{\theta}; \mathbf{p}_i, w_i) = \mathbb{1}_{f(\mathbf{p}_i) \neq w_i}.$$

Learning: Structural Support Vector Machine (SVM)

Let $S = \{(\mathbf{p}_1, w_1), \ldots, (\mathbf{p}_m, w_m)\}$. The goal is find

$$\boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta}} \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2 + \sum_{i=1}^{m} \ell(\boldsymbol{\theta}; \mathbf{p}_i, w_i),$$

where

$$\ell(\boldsymbol{\theta}; \mathbf{p}_i, w_i) = \mathbb{1}_{f(\mathbf{p}_i) \neq w_i}.$$

We cannot optimize zero-one loss directly. A common trick is to optimize the hinge loss,

$$\ell(\boldsymbol{\theta}; \mathbf{p}_i, w_i) = \max_{w \in \mathcal{V}} \left[\mathbb{1}_{w_i \neq w} - \boldsymbol{\theta}^\top \phi(\mathbf{p}_i, w_i) + \boldsymbol{\theta}^\top \phi(\mathbf{p}_i, w) \right].$$

Learning: Structural Support Vector Machine (SVM)

Let $S = \{(\mathbf{p}_1, w_1), \ldots, (\mathbf{p}_m, w_m)\}$. The goal is find

$$\boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta}} \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2 + \sum_{i=1}^{m} \ell(\boldsymbol{\theta}; \mathbf{p}_i, w_i),$$

where

$$\ell(\boldsymbol{\theta}; \mathbf{p}_i, w_i) = \mathbb{1}_{f(\mathbf{p}_i) \neq w_i}.$$

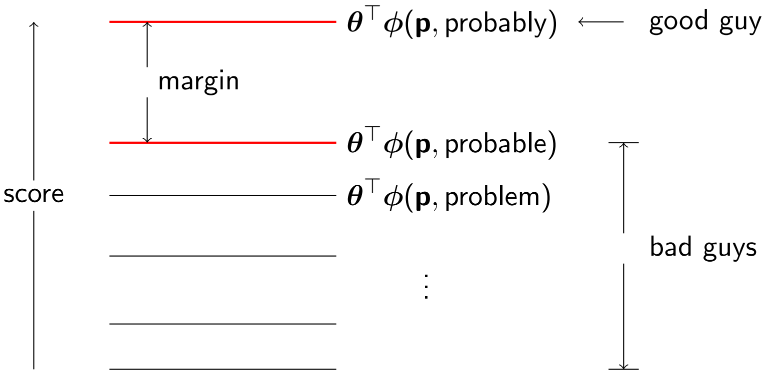
We cannot optimize zero-one loss directly. A common trick is to optimize the hinge loss,

$$\ell(\boldsymbol{\theta}; \mathbf{p}_i, w_i) = \max_{w \in \mathcal{V}} \left[\mathbb{1}_{w_i \neq w} - \boldsymbol{\theta}^\top \phi(\mathbf{p}_i, w_i) + \boldsymbol{\theta}^\top \phi(\mathbf{p}_i, w) \right].$$

We use Pegasos [Shalev-Shwartz et al., 2007] to solve the above problem.

Large-Margin Learning: Intuition

Given $\mathbf{p}$ = [pcl p r aa bcl b l iy], we want to find $\boldsymbol{\theta}$ such that



Experiments: Setting

dataset	Switchboard
lexicon	3328 words
total tokens	3344 tokens
length differences	-3, -2, -1, 0, 1, 2, 3
asynchrony	tongue tip and tongue body lip and tongue lip, tongue and glottis, velum
asynchrony degree	$(-\infty, -3), [-3, 2), [-2, -1), [-1, 0),$ $[0, 1), [1, 2), [2, 3), [3, \infty)$

Experiments: Result

Training	2942 tokens
Dev	165 tokens
Test	237 tokens

Model	Error Rate
lexicon lookup (from [Livescu, 2005])	59.3%
lexicon + Levenshtein distance	41.8%
articulatory based DBN [Jyothi et al., 2011]	29.1%
Passive-Aggressive/ALL	**15.2%**

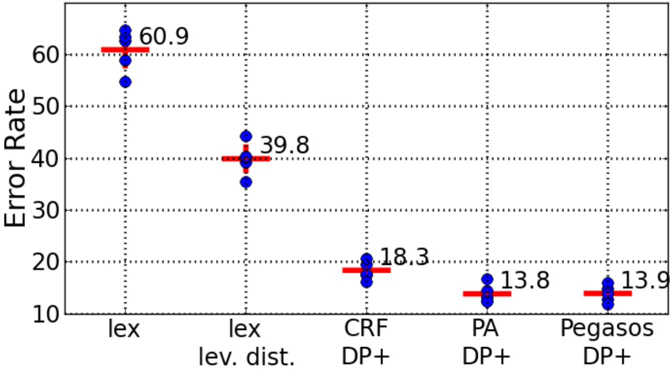
Experiments: Comparing learning methods

Algorithm	CRF	PA and Pegasos
# of non-zero entries in $\boldsymbol{\theta}$	4,000,000	800,000
Time for each epoch	45 min	15 min

DP+ dictionary, length, phone bigram TF-IDF, phonetic alignment

Experiments: Comparing learning methods

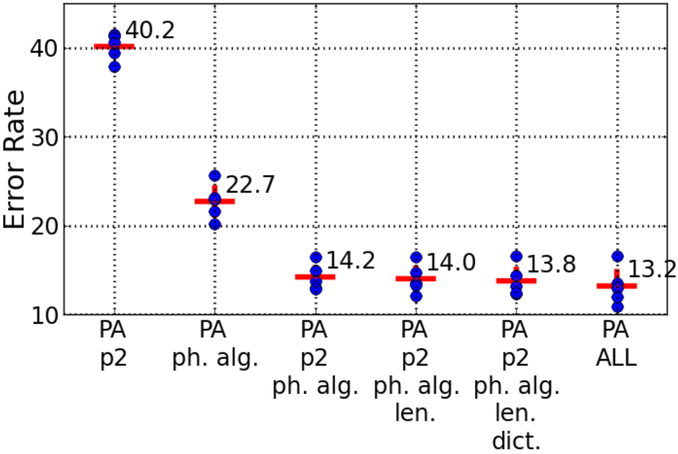
5-fold cross-validation for different learning methods.



DP+ dictionary, length, phone bigram TF-IDF, phonetic alignment

Experiments: Feature combinations

5-fold cross-validation for different feature combinations.



Example of Learned Weights

$\theta_{\mathbf{p} \in pron(w)}$	0.562960
$\theta_{\mathrm{p} \rightarrow \mathrm{p}}$	0.187971
$\theta_{\mathrm{t} \rightarrow \mathrm{dx}}$	0.291054
$\theta_{\mathrm{oy1} \rightarrow \mathrm{oy\_n1}}$	0.065720
$\theta_{\mathrm{oy2} \rightarrow \mathrm{oy\_n2}}$	0.065720
$\theta_{\mathrm{n} \rightarrow \mathrm{r}}$	-0.029258
$\theta_{\mathrm{f} \rightarrow \mathrm{kcl}}$	-0.020868
$\theta_{\Delta\ell < -3 \text{ for probably}}$	0.131365
$\theta_{\Delta\ell = -3 \text{ for probably}}$	-0.010327
$\theta_{\Delta\ell = -2 \text{ for probably}}$	0.019158
$\theta_{\Delta\ell = -1 \text{ for probably}}$	0.122276

Conclusion

- Propose a discriminative framework for pronunciation modeling
- Incorporate a large set of complex features
- Use large-margin learning

Future Work

- Acoustics
  - Align posteriors with baseforms in the dictionary
  - Extend TF-IDF to soft counts from posteriors.
- Word Sequences
  - Lattice rescoring
  - First-pass decoding
- Compare with SCRF [Zweig and Nguyen, 2009]

[th ae ng kcl k] [y uw]

Reference

K. Livescu
Feature-based Pronunciation Modeling for Automatic Speech Recognition.
Ph.D. thesis, Massachusetts Institute of Technology, 2005.

P. Jyothi, K. Livescu, and E. Fosler-Lussier
Lexical access experiments with context-dependent articulatory feature-based models.
In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2011.

G. Zweig, P. Nguyen, and A. Acero
Continuous speech recognition with a TF-IDF acoustic model.
In Proc. Interspeech, 2010.

C. P. Browman and L. Goldstein
Articulatory phonology: an overview.
Phonetica, 49(3-4), 1992.

Phonetic Alignment Feature Functions

Given $p, q \in \mathcal{P} \cup \{-\}$, we encode p and q with two four tuples (s_1, s_2, s_3, s_4) and (t_1, t_2, t_3, t_4), which represents

- consonant place
- consonant manner
- vowel place
- vowel manner.

Define the similarity between p and q as

$$s(p, q) = \begin{cases} 1, & \text{if } p = - \vee q = -; \\ \sum_{i=1}^{4} \mathbb{1}_{s_i = t_i}, & \text{otherwise,} \end{cases}$$

and run dynamic programming.

Phonetic Alignment Feature Functions

The alignment feature function for $p \to q$, for $p, q \in \mathcal{P} \cup \{-\}$, is defined as,

$$\phi_{p \to q}(\mathbf{p}, w) = \frac{1}{Z_p} \sum_{k=1}^{K_w} \sum_{i=1}^{L_k} \mathbb{1}_{a_{k,i}=p, b_{k,i}=q},$$

where $K_w = |pron(w)|$, L_k is the length of the k-th alignment, and

$$Z_p = \begin{cases} \sum_{k=1}^{K_w} \sum_{i=1}^{L_k} \mathbb{1}_{a_{k,i}=p}, & \text{if } p \in \mathcal{P}; \\ |\mathbf{p}| K_w, & \text{if } p = -. \end{cases}$$

Articulatory Feature Functions

Let $\mathcal{F}$ be the set of articulatory variables that consists of

- tongue tip location
- tongue tip opening
- tongue body location
- tongue body opening
- lip opening
- glottis
- velum

Articulatory Feature Functions

Given $p, q \in F$, for $F \in \mathcal{F}$, the feature function for articulatory alignment is defined as

$$\phi_{p \to q}(\mathbf{p}, w) = \frac{1}{L} \sum_{i=1}^{L} \mathbb{1}_{a_i = p, b_i = q}$$

Articulatory Feature Functions

surface	s	s	eh	eh_n	eh_n	n	t	s	s	s
voicing	-	-	+	+	+	+	-	-	-	-
	s	s	eh	n	n	n	s	s	s	s
nasality	-	-	-	+	+	+	-	-	-	-
	s	s	eh	n	n	n	s	s	s	s
tongue body	u	u	u	p	p	u	u	u	u	u
	s	s	eh	eh	eh	n	n	s	s	s
tongue tip	cr	cr	cr	m	m	cl	cl	cr	cr	cr
	s	s	eh	eh	eh	n	n	s	s	s
asynchrony				1	1		1			

Articulatory Feature Functions

For $F_h, F_k \in \mathcal{F}$, the asynchrony between F_h and F_k is defined as

$$\text{async}(F_h, F_k) = \frac{1}{L}\sum_{i=1}^{L}(t_{h,i} - t_{k,i})$$

More generally, for $\mathcal{F}_1, \mathcal{F}_2 \subset \mathcal{F}$, the asynchrony between $\mathcal{F}_1$ and $\mathcal{F}_2$ is defined as

$$\text{async}(\mathcal{F}_1, \mathcal{F}_2) = \frac{1}{L}\sum_{i=1}^{L}\left[\frac{1}{|\mathcal{F}_1|}\sum_{F_h \in \mathcal{F}_1} t_{h,i} - \frac{1}{|\mathcal{F}_2|}\sum_{F_k \in \mathcal{F}_2} t_{k,i}\right]$$

Define the asynchrony among articulatory variables feature functions as

$$\phi_{a \le \text{async}(\mathcal{F}_1, \mathcal{F}_2) \le b}(\mathbf{p}, w) = \mathbb{1}_{a \le \text{async}(\mathcal{F}_1, \mathcal{F}_2) \le b}$$

Experiments

Training	2942 tokens
Dev	165 tokens
Test	237 tokens

Model	ER
lexicon lookup (from [Livescu, 2005])	59.3%
lexicon + Levenshtein distance	41.8%
[Jyothi et al., 2011]	29.1%
CRF/DP+	21.5%
PA/DP+	**15.2%**
Pegasos/DP+	**14.8%**
PA/ALL	**15.2%**