

# Automatic generation of phone and state level acoustic model mappings across languages

*Presented by Juan M. Huerta*

# Introduction to cross-lingual automatic model mapping

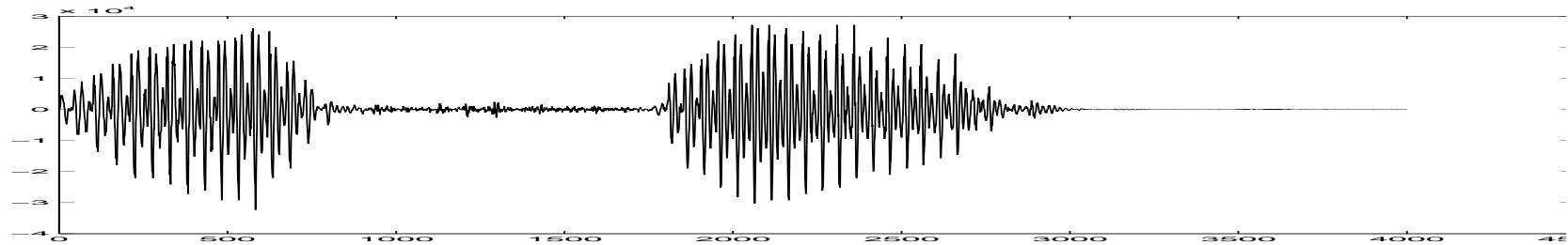
Given a collection of well trained acoustic models in several source languages we want to know:

- What is the best use we can give to these models when recognizing in a new target language?
- Are **automatic mappings** a robust modeling technique when data is really scarce?

Some use that we can give to mappings

- Data within domain is extremely scarce (few minutes)
- Use it as a *seed* model to initialize further training
- Use it as to enrich approaches like DMC, ROVER
- Borrow data across languages

# The Confusion Matrix approach for finding cross-lingual phone similarities



X:	x_a	x_f	x_aa	x_sil	
Y:	y_a	y_f	y_a	y_n	y_sil

Count the co-occurrences between outcomes of R.V.'s X and Y:

$$M(i, j) = C(x_i, y_j)$$

Normalizing the matrix, yields a joint pdf of the R.V.'s X and Y:

$$p(x_i, y_j) = \frac{C(x_i, y_j)}{\sum_x \sum_y C(x_i, y_j)}$$

## Criteria for mappings from the Confusion Matrix

1.- Given a Czech label (e.g., phone)  $x_i$ , find the model  $y_j$

$$M(x_i) = \underset{y_j}{\operatorname{argmax}} p(y_j | x_i) = \underset{y_j}{\operatorname{argmax}} \frac{C(x_i, y_j)}{C(x_i)} \Rightarrow \underset{y_j}{\operatorname{argmax}} C(x_i, y_j)$$

2.- We can maximize the posterior prob of  $x_i$  given that  $y_j$  was observed

$$M(x_i) = \underset{y_j}{\operatorname{argmax}} p(x_i | y_j) = \underset{y_j}{\operatorname{argmax}} \frac{p(y_j | x_i) p(x_i)}{p(y_j)} \Rightarrow \underset{y_j}{\operatorname{argmax}} \frac{C(x_i, y_j)}{C(y_j)}$$

3.- The map that leaves the smallest total # of counts outside the mapping region allowing any model  $y_j$  to represent only one  $x_i$ :

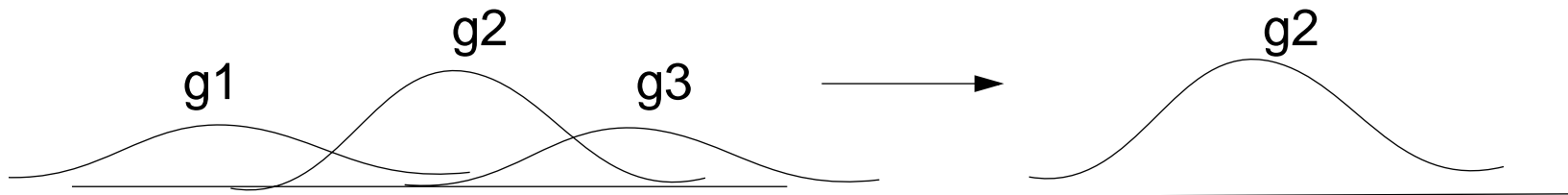
		$y_1$		$y_2$	
		1		3	
$x_1$	5	7	0	7	0
		0		3	
$x_2$	0	7	0	3	0

## From mappings to new models

Czech models are assembled as a weighted combination of the best n matching models

$$p(o|Cza\bar{1}) = \sum_i \sum_j w(i, j) c(j) N(o, m_{ij}, C_{ij})$$

Alternatively, one can select the n Gaussian with the highest mixing weights



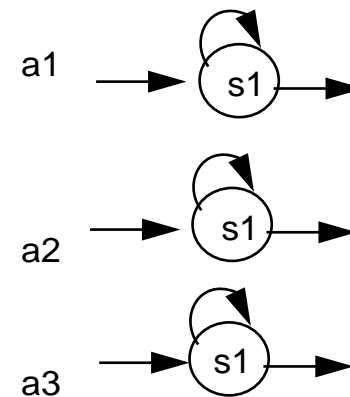
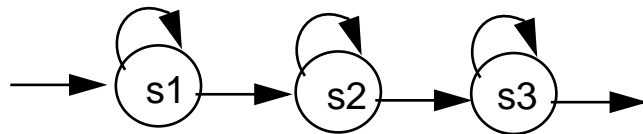
Now we are able to obtain a HMM model definition file in terms of models coming from other languages

# Generating sub-phonetic similarities: state-level Confusion Matrix

Some phones seem to be **hard to match** against in other languages:

- Use **phonetic subunits**: construct a Matrix at the state level treating every state as a separate phonetic entity

Model: "a"



- The alignments and phonetic recognition are performed now using the states separately

## State mappings, 3 best matches: English, Spanish, Mandarin, Russian

a_3	EN_aax_3	0.06	MA_at_2	0.04	SP_a_2	0.04
a_4	SP_a_3	0.05	EN_ao_3	0.04	MA_gt_3	0.04
aa_2	EN_ay_1	0.12	EN_aa_1	0.11	MA_a_1	0.09
aa_3	EN_aw_2	0.19	EN_aa_2	0.17	EN_aax_2	0.12
aa_4	EN_ay_2	0.28	EN_aw_3	0.23	EN_ao_3	0.06
aw_2	EN_ow_1	1.00	SP_y_3	0.00	SP_y_2	0.00
aw_3	SP_rr_3	0.51	MA_u_1	0.49	SP_y_3	0.00
aw_4	EN_uh_2	1.00	SP_y_3	0.00	SP_y_2	0.00
b_2	RU_b_1	0.24	RU_bj_1	0.15	EN_b_1	0.06
b_3	RU_b_2	0.16	RU_bj_2	0.14	EN_w_1	0.05
b_4	EN_b_3	0.12	EN_w_2	0.09	RU_b_3	0.09
c_2	RU_c_1	0.18	SP_x_1	0.09	MA_j_1	0.08
c_3	MA_c_2	0.18	RU_c_2	0.18	MA_x_2	0.07
c_4	RU_c_3	0.10	MA_s_3	0.10	MA_x_3	0.10
ch_2	RU_chj_1	0.28	EN_ch_1	0.27	EN_jh_1	0.10
ch_3	RU_chj_2	0.25	EN_ch_2	0.20	MA_q_2	0.17
ch_4	EN_ch_3	0.19	EN_sh_3	0.17	MA_q_3	0.13
d_2	RU_d_2	0.06	RU_dj_1	0.05	MA_d_1	0.04
d_3	SP_ll_2	0.05	MA_Z_2	0.04	RU_gj_2	0.04

d_4	EN_d_3	0.05	RU_bj_3	0.04	RU_d_3	0.04
dj_2	SP_y_1	0.13	RU_zj_2	0.13	EN_jh_1	0.11
dj_3	SP_y_2	0.28	RU_dj_2	0.13	RU_gj_2	0.11
dj_4	SP_ll_3	0.18	RU_z_3	0.10	RU_gj_3	0.10
e_2	MA_E_1	0.04	RU_ee_2	0.03	RU_e_2	0.02
e_3	MA_E_2	0.03	RU_e_3	0.02	RU_ee_3	0.02
e_4	SP_e_3	0.03	EN_n_1	0.03	MA_E_3	0.02
ee_2	EN_eh_1	0.09	EN_ao_1	0.09	EN_eh_2	0.08

Same as above, but with competing Czech models

a_3	CZ_a_2	0.06	MA_at_2	0.05	EN_aa_2	0.04
a_4	CZ_a_3	0.06	SP_a_3	0.04	MA_gt_3	0.03
aa_2	CZ_aa_1	0.25	MA_a_1	0.13	EN_ay_1	0.09
aa_3	CZ_aa_2	0.28	EN_aax_2	0.15	EN_aw_2	0.14
aa_4	CZ_aa_3	0.21	EN_aw_3	0.13	EN_ay_2	0.12
aw_2	CZ_aw_1	0.50	MA_R_1	0.32	EN_ow_1	0.15
aw_3	CZ_aw_2	0.86	MA_u_1	0.09	RU_l_1	0.05
aw_4	CZ_aw_3	0.97	EN_uh_2	0.03	SP_y_3	0.00
b_2	CZ_b_1	0.28	RU_b_1	0.16	RU_bj_1	0.08
b_3	CZ_b_2	0.28	RU_bj_2	0.13	RU_b_2	0.11
b_4	CZ_b_3	0.31	EN_w_2	0.08	EN_b_3	0.07

---

c_2	CZ_c_1	0.42	MA_c_1	0.09	MA_q_1	0.07
c_3	CZ_c_2	0.38	RU_c_2	0.10	MA_x_2	0.09
c_4	CZ_c_3	0.42	MA_c_3	0.06	SP_s_2	0.06
ch_2	CZ_ch_1	0.51	SP_ch_2	0.20	EN_ch_1	0.16
ch_3	CZ_ch_2	0.37	RU_chj_2	0.32	SP_ch_2	0.10
ch_4	CZ_ch_3	0.38	RU_shj_3	0.28	RU_sh_3	0.11
d_2	CZ_d_1	0.10	RU_dj_1	0.03	EN_g_1	0.03
d_3	CZ_d_2	0.08	RU_gj_2	0.06	SP_ll_2	0.05
d_4	CZ_d_3	0.07	RU_chj_2	0.03	CZ_g_3	0.03
dj_2	CZ_dj_1	0.41	RU_zj_2	0.16	RU_c_1	0.06
dj_3	CZ_dj_2	0.56	MA_j_3	0.12	SP_y_2	0.11
dj_4	CZ_dj_3	0.51	RU_gj_3	0.11	SP_ll_3	0.08
e_2	MA_E_1	0.03	CZ_e_1	0.03	RU_e_2	0.02
e_3	CZ_e_2	0.03	EN_ae_2	0.03	MA_E_2	0.02
e_4	CZ_e_3	0.03	SP_e_3	0.03	EN_n_1	0.02

## Experimental results

Source(s)	level	method	n-best	%WER.
<b>Czech</b>	<b>Phone</b>	<b>baseline</b>	<b>1</b>	<b>38.01%</b>
E	Phone	manual	1	>80%
S	Phone	manual	1	>80%
E	Phone	matrix	1	68.31%
S	Phone	matrix	1	68.67%
E	State	matrix	1	64.75%
S	State	matrix	1	70.03%
M	State	matrix	1	79.69%
E+S+M	State	matrix	1	62.28%
E+S+M	State	matrix	3	55.77%
E+S+M	State	matrix-2	3	54.38%

---

## Experimental Results (2)

When the czech data is used for further improvements:

- Full retrain gave practically same result as flat start 36.8% vs. 37.07% WER, partial retrain (only means etc.) was worse
- Compensation through cepstral mean normalization brings WER down to 49% (additive term) from ~55% with a very limited amount of data (~6 minutes!), details later by Bhaskara

Stuff that didn't go so smooth... :- (

- After a point adding more languages stops helping (e.g., Russian)
- Gaussian mixture selection/reduction/interpolation didn't help
- Retraining from mappings, gave practically same result as flat start...
- Baseline 1-hour system performance was so high...

## Follow-up work:

# Incorporating mapping and environmental compensation methods

Refinements to the method:

- Phonetically constrained mappings
- Extension to context dependent models (triphones)

### **New direction:**

Perform a data-normalization in the source languages before obtaining the mappings (make the source data more similar to the target language data so mappings are not heavily influenced by similar acoustic environments) and the phonetic corrections

# Incorporating mapping and environmental compensation methods

