Segment Distances and Foreign Accents

Martijn Wieling and John Nerbonne

Center for Language and Cognition Groningen, University of Groningen

LOT Winterschool 2012, Tilburg, January 12

Overview

- Segment distances
  - Why use sensitive segment distances?
  - Obtaining sensitive segment distances
  - Evaluating the quality of (using) sensitive segment distances
- English accents
  - The Speech Accent Archive
  - A visualization of English accents
  - Linking computational and perceptual pronunciation distances
  - A regression model to predict word pronunciation distances

Collaborators






Introduction

- In the previous lectures: measuring pronunciation differences
- The Levenshtein (edit) distance is central in our approach
  - A very rough measure: the minimum number of insertions, deletions and substitutions to transform one string into the other
  - No distinction between sound segment substitutions involving similar sounds from different sounds: [i]:[y] vs. [a]:[i]
- Here we will introduce an extension of the Levenshtein distance which uses (automatically derived) sensitive segment distances
- Can you think of reasons why (and when) this would be an improvement?

Introduction

- In the previous lectures: measuring pronunciation differences
- The Levenshtein (edit) distance is central in our approach
    - A very rough measure: the minimum number of insertions, deletions and substitutions to transform one string into the other
    - No distinction between sound segment substitutions involving similar sounds from different sounds: [i]:[y] vs. [a]:[i]
- Here we will introduce an extension of the Levenshtein distance which uses (automatically derived) sensitive segment distances
- Can you think of reasons why (and when) this would be an improvement?

Recap: Levenshtein distance (VC-sensitive)

mɔəlkə	delete ə	1
mɔlkə	subst. ɔ/ɛ	1
mɛlkə	delete ə	1
mɛlk	insert ə	1
mɛlək		
		4

m	ɔ	ə	l		k	ə
m	ɛ		l	ə	k	
	1	1		1		1

- Note that the alignment results in an implicit identification of sound segment correspondences

Recap: Levenshtein distance (VC-sensitive)

mɔəlkə	delete ə	1
mɔlkə	subst. ɔ/ɛ	1
mɛlkə	delete ə	1
mɛlk	insert ə	1
mɛlək		
		4

m	ɔ	ə	l		k	ə
m	ɛ		l	ə	k	
	1	1		1		1

- Note that the alignment results in an implicit identification of sound segment correspondences

Counting sound segment correspondences

- Counting the frequency of sound segments (in the alignments)

p	b	...	ʊ	u	Total
5×10^5	2×10^5	...	90,000	9×10^5	10^8

- Counting the frequency of the aligned sound segments (in the alignments)

	p	b	...	ʊ	u	
p	2×10^5	60,650	...	0	0	
b		88,000	...	0	0	
⋮			⋮	⋮	⋮	
ʊ				65,400	5,500	
u					4×10^5	
						Total: 10^7

- Probability of observing [p]: 5×10^5 / 10^8 = 0.005 (0.5%)
- Probability of observing [b]: 2×10^5 / 10^8 = 0.002 (0.2%)
- Probability of observing [p]:[b]: 60,650 / 10^7 = 0.006 (0.6%)

Association strength between sound segment pairs

- Pointwise Mutual Information (PMI): assesses degree of statistical dependence between aligned segments (x and y)

$$\mathrm{PMI}(x, y) = \log_2 \left(\frac{p(x, y)}{p(x)\, p(y)} \right)$$

- $p(x, y)$: relative occurrence of the aligned segments x and y in the whole dataset
- $p(x)$ and $p(y)$: relative occurrence of x and y in the whole dataset

- The greater the PMI value, the more sound segments tend to cooccur in correspondences

Association strength between sound segment pairs

- Probability of observing [p]:[b]: 60,650 / 10^7 = 0.006
- Probability of observing [p]: 5×10^5 / 10^8 = 0.005
- Probability of observing [b]: 2×10^5 / 10^8 = 0.002

$$\mathrm{PMI}(x,y) = \log_2 \left(\frac{p(x,y)}{p(x)\,p(y)} \right) \Rightarrow$$

$$\mathrm{PMI}(\mathrm{p},\mathrm{b}) = \log_2 \left(\frac{0.006}{0.005 \times 0.002} \right)$$

$$\mathrm{PMI}(\mathrm{p},\mathrm{b}) \approx 9.2$$

Using PMI values with the Levenshtein algorithm

- Idea: use association strength to weight edit operations
- PMI is large for strong associations, so we invert it (0 - PMI)
  - Strongly associated segments will have a low distance
- PMI range varies, so we normalize it between 0 and 1
- Use PMI-induced weights as costs in Levenshtein algorithm
  - Cost of substituting identical sound segments is always set to 0

The PMI-based Levenshtein algorithm

- We use the VC-sensitive Levenshtein algorithm to calculate the initial PMI weights and convert these to costs (i.e. sound distances)
- These sensitive sound segment distances are then used as edit operation costs in the Levenshtein algorithm to obtain new alignments, new counts, and new PMI sound distances
- This process is repeated until alignments and PMI sound segment distances stabilize

Evaluating alignment quality

- Dataset: Bulgarian dialect transcriptions (197 sites, 152 words)
- A gold standard set of 3.5 million pairwise alignments was used for evaluation (automatically generated from a multiple alignment)
- We compare the VC-sensitive Levenshtein algorithm with the PMI-based Levenshtein algorithm
  - We also evaluate a slightly modified version of the PMI-based Levenshtein algorithm where we exclude identical sound segment substitutions from all counts (diagonal-exclusive version)

Evaluation procedure (1)

- The pairwise alignments are generated by the algorithms
  - Insertion-deletion sequences are standardized:

v	'i		ɑ	|	v	'i	ɑ	
v	'i	j		|	v	'i		j

  - Two-to-one mappings are standardized:

v		'ɹ̩	x	|	v	'ɹ̩		x
v	'ɑ	r	x	|	v	'ɑ	r	x

Evaluation procedure (2)

- Each sound segment alignment is converted to a single symbol:

v	l	ˈɣ	k	v	l	ˈɣ		k
v	ˈɣ	l	k	v		ˈɣ	l	k
v/v	l/ˈɣ	ˈɣ/l	k/k	v/v	l/-	ˈɣ/ˈɣ	-/l	k/k

- These can be aligned to determine their distance:

v/v	l/ˈɣ	ˈɣ/l		k/k
v/v	l/-	ˈɣ/ˈɣ	-/l	k/k
	1	1	1	

Evaluation procedure (3)

- For all algorithms the generated strings (representing alignments) are aligned with the generated strings of the gold standard (GS)
- The total error of each algorithm is the sum of all differences with respect to the GS (based on 3.5 million word alignments, and 16 million sound segment alignments)

Alignment quality improves significantly

	Segment errors	Alignment errors
Baseline (Hamming)	2,510,094 (15.81%)	726,844 (20.92%)
Levenshtein VC	490,703 (3.09%)	191,674 (5.52%)
Levenshtein PMI	399,216 (2.51%)	156,440 (4.50%)
Levenshtein PMI (DE)	387,488 (2.44%)	152,808 (4.40%)

Example of the improvements

VC-sensitive Levenshtein algorithm, two possibilities:

b		ɪ	n	d	ə	n
b	ɛ	i	n	d	ə	
	1	1				1

b	ɪ		n	d	ə	n
b	ɛ	i	n	d	ə	
	1	1				1

PMI-based Levenshtein algorithm, only one:

b		ɪ	n	d	ə	n
b	ɛ	i	n	d	ə	
	0.034	0.020				0.024

Evaluating sound segment quality

- Besides focusing on the quality of the alignments, we can also investigate the quality of the underlying PMI-based sound segment distances
- In the following, we will show how well the automatically obtained PMI-based sound segment distances match acoustic distances (for vowels)

Pronunciation data

- Six independent dialect data sets (IPA pronunciations)
  - Dutch: 562 words in 613 locations (Wieling et al., 2007)
  - German: 201 words in 186 locations (Nerbonne and Siedle, 2005)
  - U.S. English: 153 words in 483 locations (Kretzschmar, 1994)
  - Bantu (Gabon): 160 words in 53 locations (Alewijnse et al., 2007)
  - Bulgarian: 152 words in 197 locations (Prokić et al., 2009)
  - Tuscan: 444 words in 213 locations (Montemagni et al., in press)
- For all datasets sound segment distances are obtained using the PMI-based Levenshtein algorithm (diagonal-exclusive version)

Acoustic data

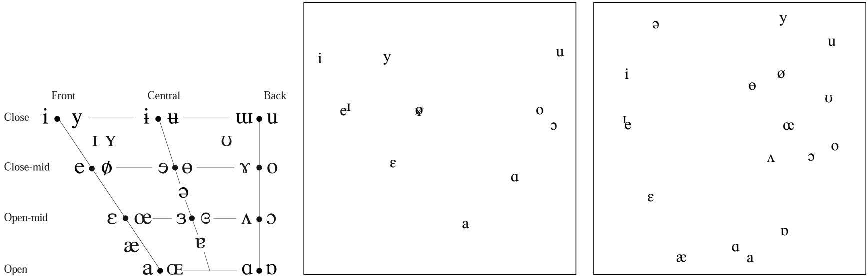
- For the evaluation, we obtained acoustic vowel measurements (F1 and F2) reported in the scientific literature
  - Pols et al. (1973; NL), van Nierop et al. (1973; NL), Sendlmeier and Seebode (2006; GER), Hillenbrand et al. (1995; US), Nurse and Phillipson (2003, p. 22; BAN), Lehiste and Popov (1970; BUL), Calamai (2003; TUS)
- To determine acoustic vowel distance, we calculate the Euclidean distance of the formant frequencies
  - Our perception of frequency is non-linear and calculating the Euclidean distance on the basis of Hertz values would not weigh the first formant enough
  - We therefore first scale the Hertz frequencies to Bark

Method of comparison

- We visualize the relative positions of the sound segments by applying multidimensional scaling (MDS) to the distance matrices
  - Missing distances are not allowed in the (classical) MDS procedure, so in some cases not all sound segments are visualized
- We assess the relation between the generated and acoustic distances using the Pearson correlation

MDS visualization of Dutch vowels

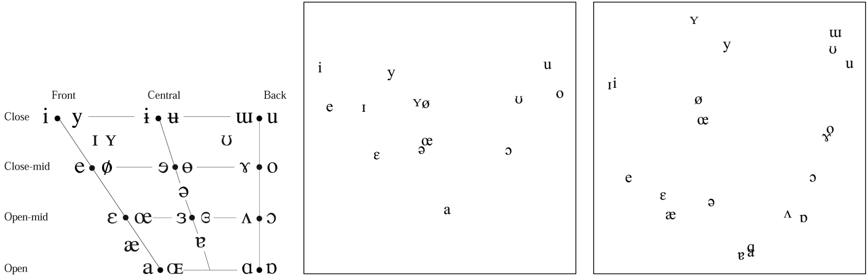
PMI visualization captures 76% of the variation



(a) IPA (b) Acoustics (c) PMI distances

MDS visualization of German vowels

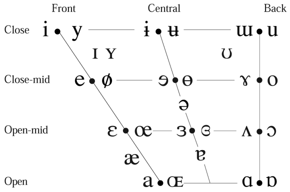
PMI visualization captures 70% of the variation



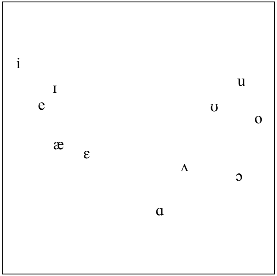
(a) IPA (b) Acoustics (c) PMI distances

MDS visualization of U.S. English vowels

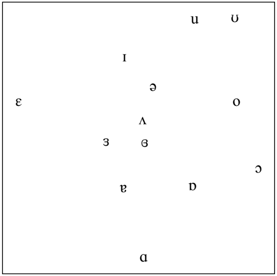
PMI visualization captures 65% of the variation



(a) IPA



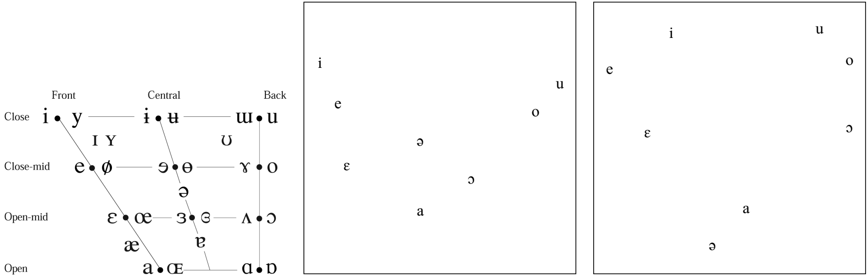
(b) Acoustics



(c) PMI distances

MDS visualization of Bantu vowels

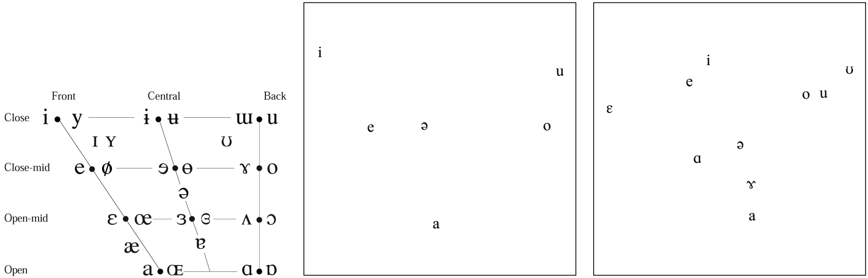
PMI visualization captures 90% of the variation



(a) IPA (b) Acoustics (c) PMI distances

MDS visualization of Bulgarian vowels

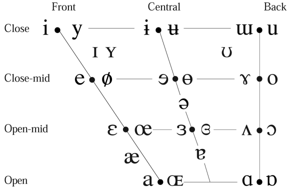
PMI visualization captures 86% of the variation



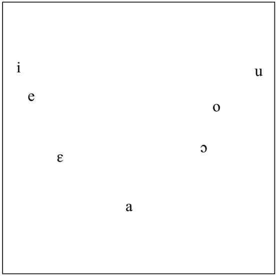
(a) IPA (b) Acoustics (c) PMI distances

MDS visualization of Tuscan vowels

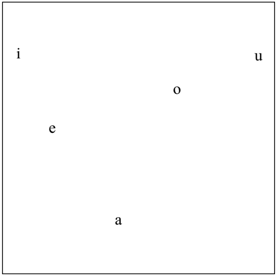
PMI visualization captures 97% of the variation



(a) IPA



(b) Acoustics



(c) PMI distances

Acoustic vs. PMI vowel distances

	Pearson's r	Explained variance (r^2)
Dutch	0.672	45.2%
Dutch w/o Frisian	0.686	47.1%
German	0.630	39.7%
German w/o ə	0.785	61.6%
US English	0.608	37.0%
Bantu	0.642	41.2%
Bulgarian	0.677	45.8%
Tuscan	0.758	57.5%

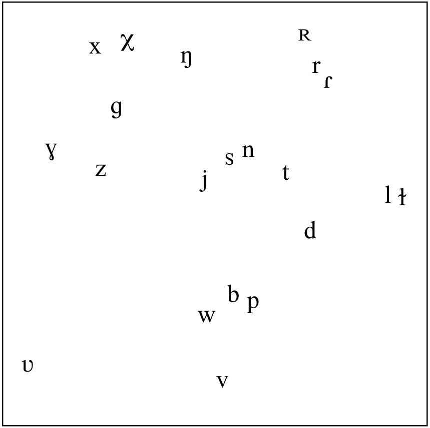
What about consonants?

- Induced distances correlate strongly with acoustic vowel distances
  - Causation is probably the reverse: acoustics explains distributions Sweeney's insight: "I gotta use words when I talk to you..."

- But for other segments (consonants) acoustic/phonetic distances are *not* well accepted, and this procedure provides a measure of distance

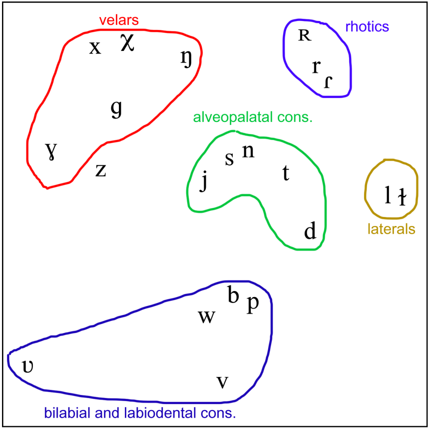
MDS visualization of Dutch consonants

PMI visualization captures 50% of the variation



MDS visualization of Dutch consonants

Place (3 groups) dominates over manner (2 groups) and voicing (no groups)



Conclusions of Part I

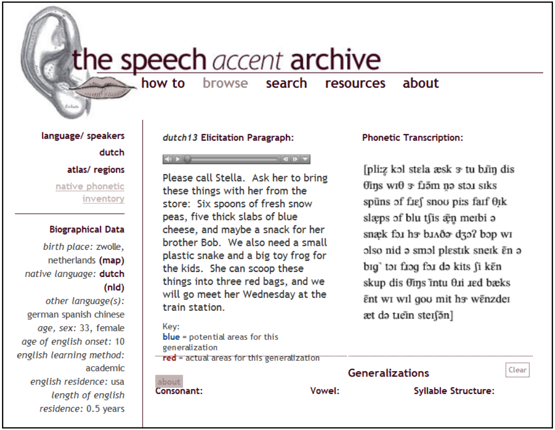
- We have shown that the PMI-based Levenshtein algorithm generates improved alignments and uses sensible sound distances
  - The approach is readily applicable to any (dialect) pronunciation dataset
- In Part II of this lecture we will apply this algorithm to obtain pronunciation distances on the basis of English Accent data
- More details (see http://www.martijnwieling.nl):
  - Martijn Wieling, Eliza Margaretha and John Nerbonne (2012). Inducing a measure of phonetic similarity from pronunciation variation. *Journal of Phonetics*, doi:10.1016/j.wocn.2011.12.004.
  - Martijn Wieling, Eliza Margaretha and John Nerbonne (2011). Inducing phonetic distances from dialect variation. *Computational Linguistics in the Netherlands Journal*, 1, 109-118.
  - Martijn Wieling, Jelena Prokić and John Nerbonne (2009). Evaluating the pairwise string alignment of pronunciations. In: Lars Borin and Piroska Lendvai (eds.) *Language Technology and Resources for Cultural Heritage, Social Sciences, Humanities, and Education (LaTeCH - SHELT&R 2009)* Workshop at the 12th Meeting of the European Chapter of the Association for Computational Linguistics. Athens, 30 March 2009, pp. 26-34

Time for a break!

The Speech Accent Archive

Available online at http://accent.gmu.edu

Audio example



Listen to an example

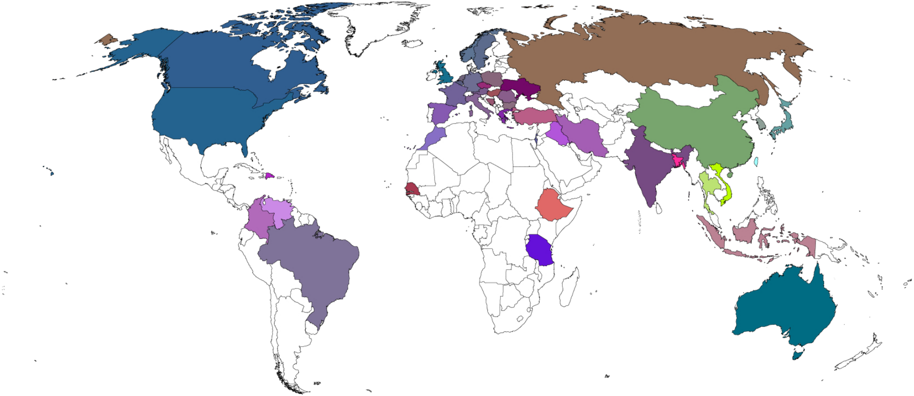
Visualizing English accents

- We used 989 phonetically transcribed samples from the SAA
- We grouped the transcriptions (i.e. speakers) per country
- For non-English speaking countries, we excluded speakers who moved to an English-speaking country before age 13
- We only included countries with at least 5 speakers

- Pronunciation distances between countries were calculated using the VC-sensitive and PMI-based Levenshtein algorithms and visualized using MDS

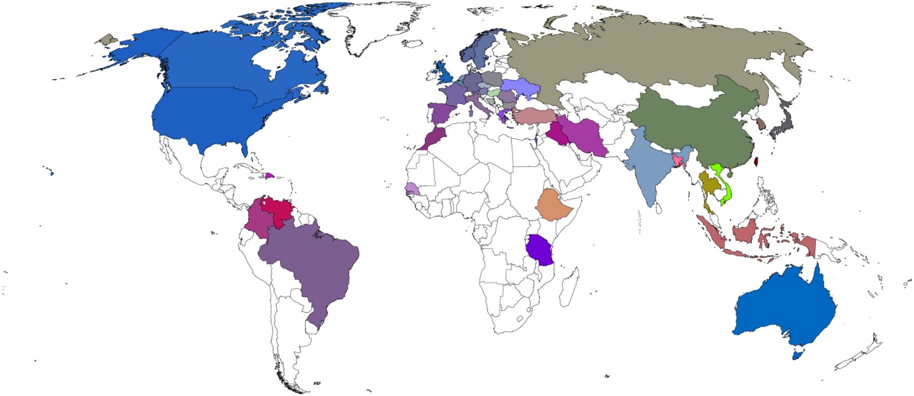
MDS visualization of accent distances

Based on the PMI-based Levenshtein algorithm (88% visualized)



MDS visualization of accent distances

Based on the VC-sensitive Levenshtein algorithm (86% visualized)



Computational vs. perceptual pronunciation distances

- There is only a single study investigating the relation between Levenshtein distances and perceptual distances
  - Focusing on Norwegian dialects (discussed on Tuesday)
  - The reported correlation strength was $r \approx 0.7$
- We conducted a new study based on the Speech Accent Archive, investigating the relation between perceptual and Levenshtein pronunciation distances
  - To illustrate this study, we will first conduct a small classroom experiment

A classroom experiment

- You will hear 4 sound samples, please rate how native-like (with respect to U.S. English) each is on a scale from 1 (very foreign sounding) to 7 (native English speaker)
  - Please write your scores down!
  - If you can, also guess the country of the speaker

Sample 1 Sample 2 Sample 3 Sample 4

What are the average classroom scores?

Levenshtein's scores

(1: very foreign sounding; 7: native English speaker)

	VC-sensitive	PMI-based
Sample 1: German speaker	4.4	4.7
Sample 2: Native U.S. speaker	7	7
Sample 3: Indonesian speaker	1.7	2.5
Sample 4: French speaker	3.4	3.6

Outline of the perception experiment

- We asked participants to answer several questions about 10 randomly selected audio samples (out of a set of 50)
  - Here we only focus on the nativeness scores
  - The samples consisted of accented speech of randomly selected male and female speakers from 26 countries
  - 89 participants filled in a questionnaire (fully or partially)
- We only included judgements of participants who were most familiar with the U.S. English variety (as opposed to U.K. English)
  - We obtained 349 nativeness scores (about 6 per sample)
- We used the Levenshtein algorithms to obtain the pronunciation distances for each of the 50 speakers and the average U.S. speaker (based on 119 samples)

Results of the perception experiment

- Corr. with the VC-sensitive Levenshtein algorithm: $r = -0.722$
- Corr. with the PMI-based Levenshtein algorithm: $r = -0.705$
- These differences are not significant
- Again, we find almost no differences between the two approaches
  - Caused by the strong similarity between the two sets of Levenshtein distances ($r^2 > 0.95$)
- But why is this happening?

Results of the perception experiment

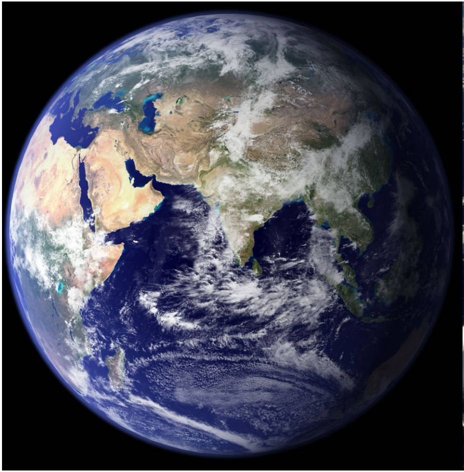
- Corr. with the VC-sensitive Levenshtein algorithm: $r = -0.722$
- Corr. with the PMI-based Levenshtein algorithm: $r = -0.705$
- These differences are not significant
- Again, we find almost no differences between the two approaches
  - Caused by the strong similarity between the two sets of Levenshtein distances ($r^2 > 0.95$)
- But why is this happening?

Results of the perception experiment

- Corr. with the VC-sensitive Levenshtein algorithm: $r = -0.722$
- Corr. with the PMI-based Levenshtein algorithm: $r = -0.705$
- These differences are not significant
- Again, we find almost no differences between the two approaches
  - Caused by the strong similarity between the two sets of Levenshtein distances ($r^2 > 0.95$)
- But why is this happening?

The level at which we compare is too high!

Sensitive segment distances do not matter when aggregating over multiple words

When are sensitive segment distances useful?

- In contrast to aggregating over multiple words, we may also look at individual word pronunciation distances
  - We already observed that alignment quality improves when using sensitive sound segment distances
  - Presumably word pronunciation distances will also improve

- In the following we will investigate which factors influence pronunciation distances from standard U.S. English speech for individual words from standard U.S. English speech

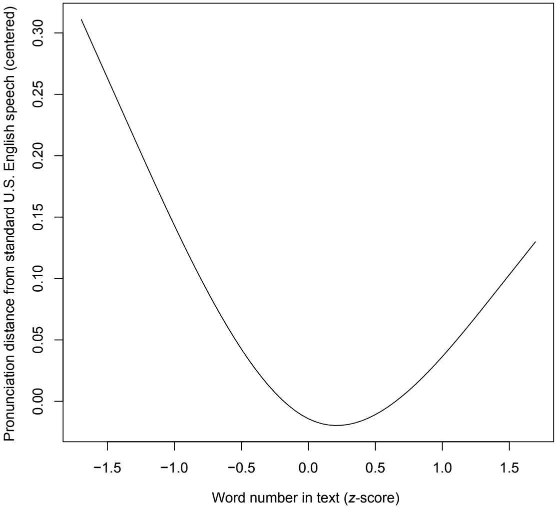
Predicting individual word pronunciation distances

- We use the PMI-based Levenshtein algorithm to obtain the pron. distances from standard U.S. English (per speaker and word)
  - We transcribed the standard U.S. English pronunciations ourselves
- We restrict our analysis to non-English speaking countries having at least 5 speakers who did not move to an English-speaking country before age 13
  - Our dataset consists of 40.000 word pronunciation distances
- We investigate the effect of several speaker, word- and country-related factors
  - We use a mixed-effects regression approach in order to take the structural variability of words, and speakers, etc. into account
  - This approach has successfully been applied to Dutch, Catalan and Tuscan dialects

Factors influencing U.S. English pron. distance

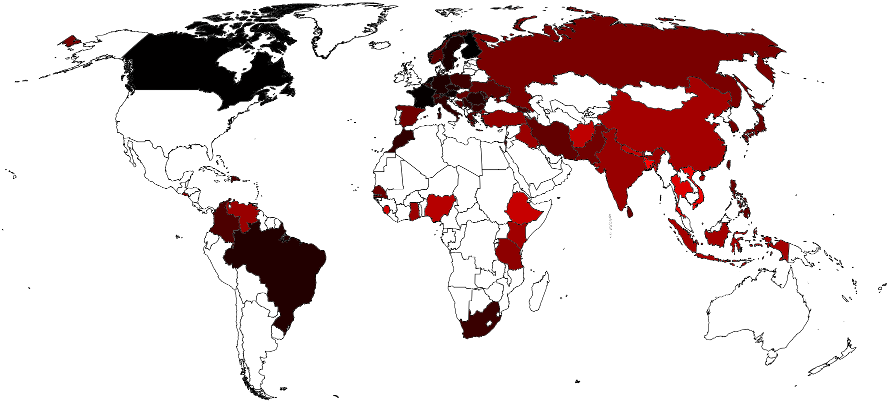
Predictor	Estimate	*t*-value
Age of English onset (log)	0.27993	10.053
Number of other languages spoken	-0.02753	-2.572
Perc. of life in English-speaking country	-0.07480	-2.932
Relative Gross Domestic Product (log)	-0.10719	-6.533
Population size (log)	0.05495	3.426
Word frequency (log)	0.14048	1.775
rcs(Word number)	-0.24428	-8.390
rcs(Word number)'	0.25447	7.128

Accents fluctuate in time



Accents compared to U.S. English speech

Structural variability of countries



Conclusions of Part II

- We have discussed several studies investigating the Speech Accent Archive
  - These studies illustrated where using sensitive sound segment distances may help and where it is not necessary
  - The results reported here are still preliminary, as the analysis of this dataset is still in progress
- More information about mixed-effects regression in dialectology (see http://www.martijnwieling.nl):

  - Martijn Wieling, John Nerbonne and R. Harald Baayen (2011). Quantitative Social Dialectology: Explaining Linguistic Variation Geographically and Socially. *PLoS ONE*, 6(9): e23613. doi:10.1371/journal.pone.0023613.
  - Martijn Wieling, Esteve Valls, R. Harald Baayen and John Nerbonne (submitted). The effects of language policies on standardization of catalan dialects: A sociolinguistic analysis using generalized additive mixed-effects regression modelling.
  - Martijn Wieling, Simonetta Montemagni, John Nerbonne and R. Harald Baayen (submitted). Lexical Differences between Tuscan Dialects and Standard Italian: A Sociolinguistic Analysis using Generalized Additive Mixed Modeling.

Thank you for your attention!