Computational Phonology, class 1: Introduction

Adam Albright

CreteLing 2022 — July 2022

SCAN ME

creteling2022.computational.phonology.party

Why computational phonology?

The many faces of computational phonology

- An engineering problem
- Distinct theoretical subfield
- Component of theoretical phonology

An engineering problem

As an engineering problem

- Encode allophony or co-occurrence restrictions, to improve accuracy or streamline TTS, speech recognition, spell-checking, etc.
- Encode variable processes (casual speech reduction, etc.)
- Characterize properties of different dialects and languages for automated language recognition

Theoretical questions

Provide guarantees about correctness or computational complexity of theories of phonological derivation or learning

- “The Recursive Constraint Demotion algorithm provably converges to a grammar that is consistent with the data” (Tesar & Smolensky 2000)
- “Gradual promotion and demotion of constraints provably converges to a grammar that is consistent with the data” (Magri 2013)
- “Phonological processes involve subregular computational complexity (Tier-based Strictly Local)” (Heinz, Chandlee, Jardine, etc.)

Theoretical questions

Highlight potential issues

- “Generating outputs in OT is NP-hard”

  (Eisner 1997; Idsardi 2006 *LI* squib and subsequent exchange with Kornai; followup by Riggle and colleagues)

- “The formal power needed to express correspondence constraints may ultimately make their evaluation intractable”

  (Potts and Pullum 2002 *Phonology* paper)

Theoretical questions

Or something in between…

- “Robust Interpretive Parsing with Resampling (RRIP) allows a learner to converge on the correct stress grammar 84% of the time using Optimality Theory with gradual constraint promotion/demotion” (Jarosz 2013)

The many faces of computational phonology

Such work is *computational* in the sense that it involves formalizing the theory in such a way that we can evaluate its computational properties. Actually *implementing* the algorithm is not necessarily instructive for these purposes.

The many faces of computational phonology

Find faster/more efficient ways to perform phonological derivations or learning

- Finite state formalization (e.g., Eisner 1999; Riggle 2004 UCLA dissertation)
- Heuristic search methods
  - Genetic algorithms (Belz and Eskikaya 1998)
  - Simulated annealing: (Bíró 2006 Groningen diss.)

The many faces of computational phonology

Find faster/more efficient ways to perform phonological derivations or learning

- Finite state formalization (e.g., Eisner 1999; Riggle 2004 UCLA dissertation)
- Heuristic search methods
  - Genetic algorithms (Belz and Eskikaya 1998)
  - Simulated annealing: (Bíró 2006 Groningen diss.)

General thrust: replace costly or uncertain procedures with ones that are more efficient, or whose properties can at least be evaluated. (And hope that the theory can still capture more or less the same facts.)

How do computational results inform theory?

- We better if we know we’re working with a system that makes phonology tractable
  - Prudent to start with mathematically well-defined models, keeping them as simple as possible
  - Elaborate slowly & cautiously to improve empirical coverage

How do computational results inform theory?

- We better if we know we're working with a system that makes phonology tractable
  - Prudent to start with mathematically well-defined models, keeping them as simple as possible
  - Elaborate slowly & cautiously to improve empirical coverage
- Yet the field "at large" often ignores such issues
  - Pragmatic approach: theory must also accommodate huge piles of complicated and diverse data
  - Often useful to pursue insights without knowing exactly how to formalize them $\Rightarrow$ preliminary sense of the fit to the data
  - If they work out well, hope that an elegant and tractable formalization can constructed post hoc

The many faces of computational phonology

Computational models as tools for developing theories

- Prosthetic thought[1]
  - Testing out ideas that would be impractical or impossible to check by hand
- Analytical hygiene
  - Verifying that a model derives the claimed output
- Baselines for evaluation
  - Qualitative and quantitative comparison of models with/without a given assumption
- Modular theorizing
  - Easily compare models that incorporate different assumptions while holding all else equal

[1] Term borrowed from Bruce Hayes.

Goals of this class

- Introduction to some basic computational concepts and approaches that are allowing phonologists to better understand humans and theories
- Demystify modeling
- I do not assume
  - Prior experience with programming or computational modeling
  - Prior knowledge of Optimality Theory, Harmonic Grammar, Maximum Entropy models, etc.
- I do assume
  - Some familiarity with phonological terminology and concepts (contrast, features, etc.)

Modeling humans

Hockett (1955) *Manual of Phonology*

- p. 147 “We know of no set of procedures by which a Martian, or a machine, could analyze a phonologic system—an entity, that is, to which even the basic biologic and cultural common denominator of humanness would be alien and would require specification. The only procedures which can be described are rules for a human investigator, and depend essentially on his ability to empathize.”

Hockett (1955) *Manual of Phonology*

- p. 147 “We know of no set of procedures by which a Martian, or a machine, could analyze a phonologic system—an entity, that is, to which even the basic biologic and cultural common denominator of humanness would be alien and would require specification. The only procedures which can be described are rules for a human investigator, and depend essentially on his ability to empathize.”
- The challenge: what do we need to build into a model in order to simulate phonological empathy?

Simulating humans

- Goal: use computational models to shed light on the phonological knowledge that humans have, and how they got it
  - Innate knowledge
  - Acquired knowledge
- Assessing our models
  - Ability to replicate the training data
  - Ability to simulate human judgments and productions (Turing test)

How do humans demonstrate phonological knowledge?

- Judgments of identity and difference (contrast)
  - Hockett (1955): American English *wood* and *would* have the same vowel, *wood* and *wooed* have different vowels
- Phonotactic acceptability
  - Chomsky and Halle (1965): *blick* [blɪk] is a possible word of English, *bnick* [bnɪk] is not
- Alternations
  - Halle (1978): English plural *Bach*-[s] (cf. *Jarre*-[z])

We'll focus in this class on modeling phonotactics and alternations

The *blick* test

Halle (1978) “Knowledge unlearned and untaught: What speakers know about the sounds of their language.”

- Which of the following could be words of English?

ptæk	plæst	vlæs
θoʊl	sram	flɪtʃ
hlad	mgla	rtut

- What is the probability that [ptæk] could be a word of English?
  - Wordlikeness judgments
- Related, but distinct: how acceptable is [ptæk]?
  - Acceptability judgments

Evidence that infants learn about inventories

Jusczyk et al. (1993) Infants' sensitivity to the sound pattern of native language words

- Experiment
  - Lists: 15 abstract low frequency words each
  - Experiment: headturn preference procedure, American 6-month olds and 9-month olds
    - Infant seated on caregiver's lap in booth
    - Green light directly ahead, "centers" infant's attention
    - Then red light on left or right blinks, and sound comes from speaker on that side
    - Trial ends when infant looks away for more than 2 secs
- Result: 9-month olds (but not 6-month olds) listen longer to English words than to Dutch words
- Many possible ways of discriminating: sounds, sound combinations, words (unlikely)

N-grams

A very simple model

- As infants are exposed to speech, they accumulate evidence about the relative frequency of sounds
- By 9 months of age, infants have collected enough statistics about English to recognize that [θ] occurs, but [x] does not
- Task behavior: look at loudspeaker in proportion to frequency of sounds coming through (≈ ‘familiarity’)

Of course, there are many other possible interpretations, too! This is merely a simple statistical baseline model.

A warm-up model: unigrams

- Count the frequency of English phones
- Calculate probability of each phone in the English lexicon
- Generalization: probability of a word is joint probability (=product of probabilities) of its phones
- Is this enough to distinguish English from Dutch, for an English-learning child?

Bigram probability for sequences

Jusczyk et al. (1993), continued… (Exp 4)

- Are 9-month olds going beyond inventories, and also learning about combinatory possibilities?
- Test: lists of words restricted to sounds that (roughly) occur in both languages, but with combinatoric violations
  - English-only combinations: final voiced obstruents (*kudos*, *cubeb*, *aboard*), word-initial schwa (*astound*)
  - Dutch-only combinations: initial [kn] (*knoest*), initial [zw] (*zweten*), [lmp] (*zalmpjes*)

Bigram probability for sequences (*cont.*)

- Tested American and Dutch infants, 6 and 9 months old
- Results
  - American 9-mo. olds listen significantly longer to English words
  - Dutch infants listen slightly (but not significantly) longer to Dutch words
  - Post hoc survey revealed that Dutch infants hear (on avg.) 1.25 hrs of English per day
  - 6 month olds don’t show same preference

Distinguish low vs. high bigram frequency

Some possibly confirming evidence: Jusczyk et al. (1994) Infants' sensitivity to phonotactic patterns in the native language

- Refinement to previous finding: common vs. rare English sequences
- High probability words
  - [rɪs], [gɛn], [kæz], [ʃæn], [sɛtʃ]
- Low probability words
  - [jaʊdʒ], [ʃɔtʃ], [ðʌʃ], [fuv], [θɔʃ]
- Tested 6- and 9-month old, as before
- Results: 9-month olds (but not 6-month olds) attend longer to high probability words

Counting combinations

The simplest type of combinations: n-grams

- Substrings of letters/segments/etc.
  - N-grams = substrings of length N
  - E.g., bigrams: *aardvark* [aɹdvaɹk] → [aɹ], [ɹd], [dv], …

Bigram probability

- Bigram frequency $=$ Count of bigram in corpus
- Probability $= \frac{\text{Count of bigram in corpus}}{\text{Count of all bigrams in corpus}}$
- Problem: how to combine probabilities of bigrams within a word to yield a probability or score for the entire word?

Bigram probability

Combining bigram frequencies into scores for an entire word (*abcd*)

- Common move in the psycholinguistics literature: either *joint* or *average* bigram probability
  - Calculate prob. of component bigrams (*ab*, *bc*, *cd*)
  - Multiply them (joint probability) or average them (average probability)

Bigram probability

Combining bigram frequencies into scores for an entire word (*abcd*)

- Common move in the psycholinguistics literature: either *joint* or *average* bigram probability
  - Calculate prob. of component bigrams (*ab*, *bc*, *cd*)
  - Multiply them (joint probability) or average them (average probability)
- However, the bigrams of a word are not independent of one another!!
  - If bigram 1 *ab* ends in *b*, bigram 2 must begin with *b*
  - i.e., by the time we get to bigram 2, the *b* is “given”
  - It makes sense at each point to focus on probability, given the current segment, that the upcoming segment should occur next

Conditional/transitional probability

- Conditional probability of *b* given *a*
  - P(b|a) = P(ab) / P(a)
- N-gram probability of string *abcd* = joint conditional probability of each segment based on preceding N-1 segments
  - Bigrams: P(*abcd*) = P(a) × P(b|a) × P(c|b) × P(d|c)
- In practice, usually sensible to avoid a word/sentence including boundaries: *#abcd#*

Observing bigram probability

- Jusczyk et al (1994): “high” vs. “low” probability nonce words
  - Are they distinguished by unigram probability?
  - Are they distinguished by transitional bigram probability?

(See Google Colab worksheet for a test)

The *blick* test in the lab

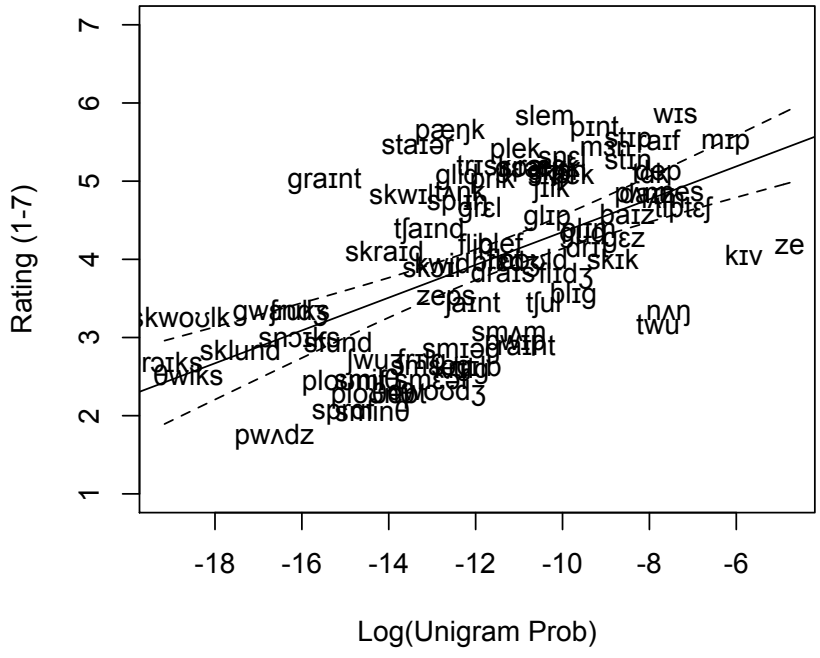
Albright and Hayes (2003): norming data for wug test of English past tenses

- Ultimate goal of study: assess acceptability of different past tense processes for nonce verbs of various shapes
  - *John likes to stin. Yesterday he stinned.*
- In theory, reactions to potential past tense forms could be influenced by two distinct factors
  - Morphological acceptability of regular -ed with stin (as opposed to stan, stun, …)
  - Phonotactic acceptability of the root (if stin is odd, then stinned will be too)
- Norming pre-test: collect ratings of phonotactic acceptability of the stems
  - How plausible would stin be as a word of English?

The *blick* test in the lab

slem	5.8	tɑk	5.1	gud	4.3	gwɛndʒ	3.3	smilθ	2.5
wɪs	5.8	tʃek	5.1	blef	4.2	ʃruks	3.3	ploʊmf	2.4
pɪnt	5.7	glid	5.1	gɛz	4.2	nʌŋ	3.3	dwoʊdʒ	2.3
pæŋk	5.6	graɪnt	5.0	drɪt	4.2	skwoʊlk	3.3	ploʊnθ	2.3
raɪf	5.5	prik	5.0	flip	4.2	twu	3.2	θept	2.3
stɪp	5.5	ʃɪlk	4.9	ze	4.2	smʌm	3.1	sminθ	2.1
mɪp	5.5	daɪz	4.8	skraɪd	4.1	snɔɪks	3.0	sprɑf	2.1
staɪər	5.5	nes	4.8	kɪv	4.1	sfund	2.9	pwʌdz	1.7
mɜn	5.4	tʌŋk	4.8	flɛt	4.0	pwɪp	2.9		
plek	5.4	skwɪl	4.8	noʊld	4.0	raɪnt	2.9		
snɛl	5.3	lʌm	4.8	skɪk	4.0	sklund	2.8		
stɪn	5.3	pʌm	4.8	brɛdʒ	3.9	smɪəg	2.8		
ræsk	5.2	splɪŋ	4.7	kwid	3.9	frɪlg	2.7		
trɪsk	5.2	grɛl	4.6	skɔɪl	3.9	ʃwuʒ	2.7		
spæk	5.2	tip	4.6	draɪs	3.8	θrɔɪks	2.7		
dep	5.1	tɛʃ	4.6	flɪdʒ	3.8	trɪlb	2.6		
gɛər	5.1	baɪz	4.6	blɪg	3.5	krɪlg	2.6		
glɪt	5.1	glɪp	4.5	zeps	3.5	smɛəg	2.6		
ʃɜn	5.1	tʃaɪnd	4.4	tʃul	3.4	θwiks	2.5		
skɛl	5.1	plɪm	4.4	ʃaɪnt	3.4	smɛrf	2.5		

Unigram probabilities (r = .575)

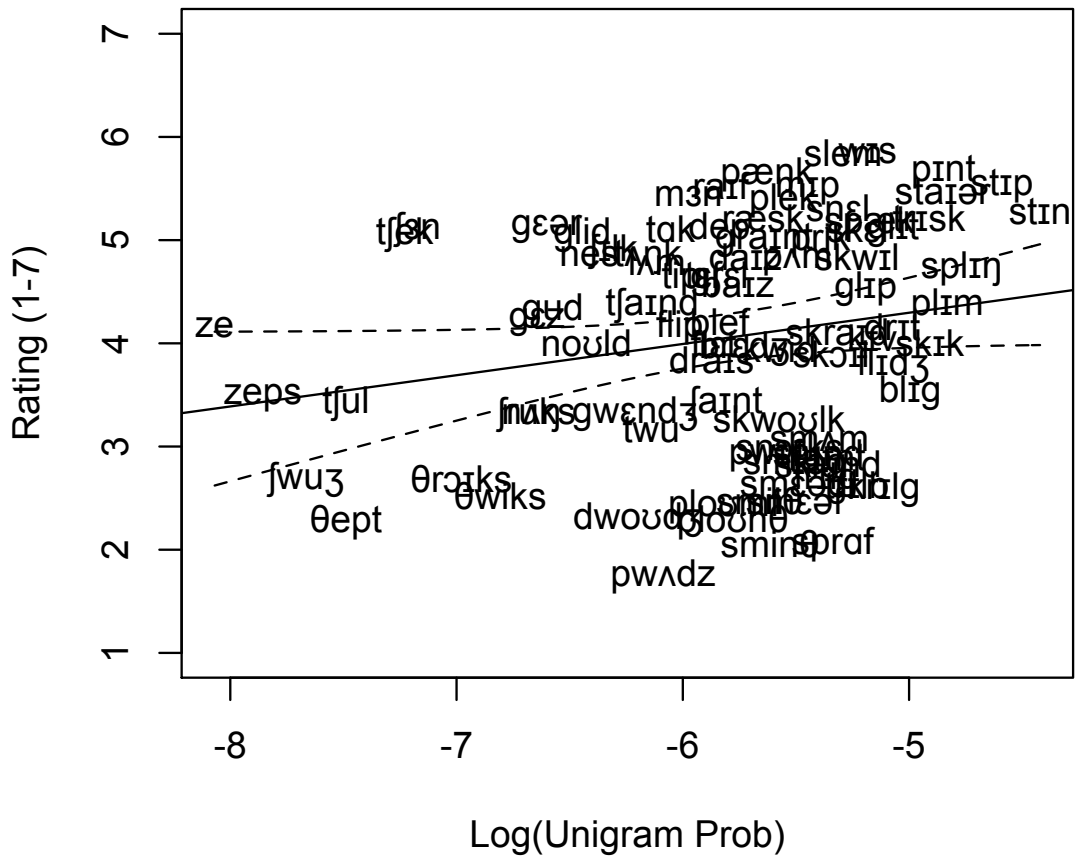


- Unigram probabilities alone already do fairly well
- Phonotactic restrictions ⇒ rare/unattested combinations ⇒ lower segment frequency

Thought experiment

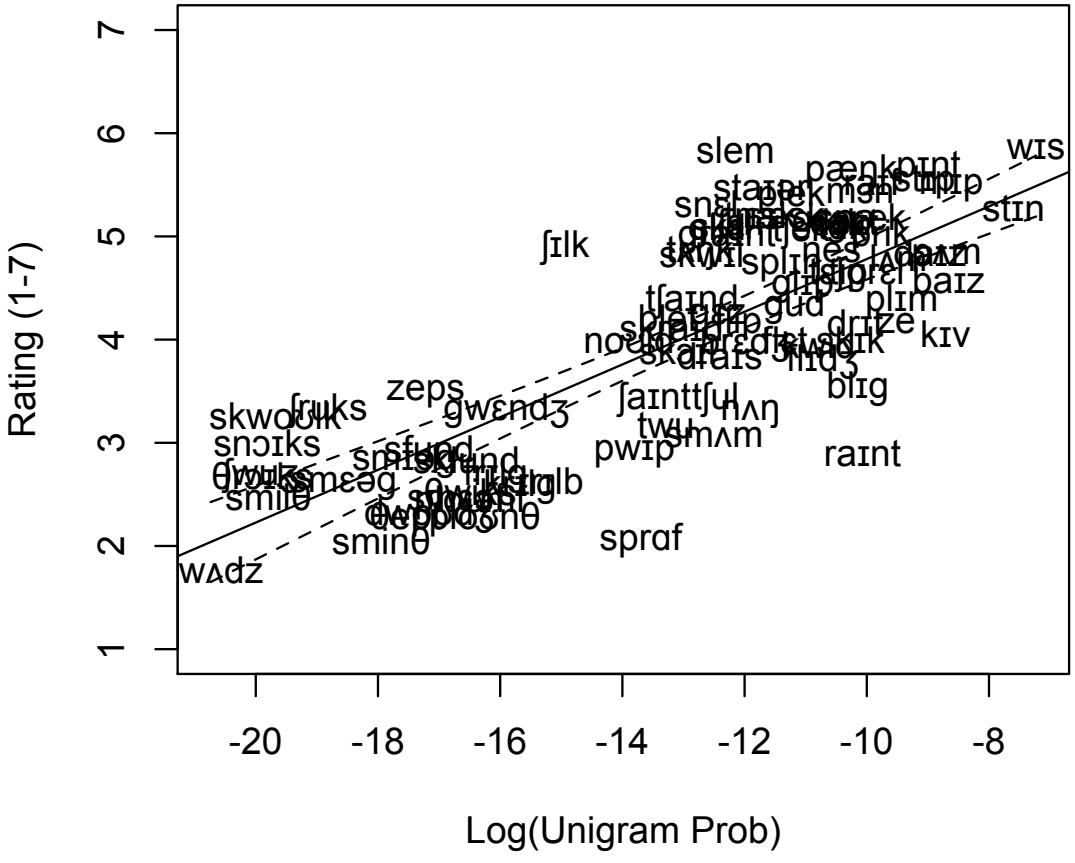
- Propose an experiment to show that unigram frequency is insufficient as a model of human judgments

Average bigram probability (r = .203)



- Averaging bigram probabilities is a much worse model
- Word length effects
- “Bottleneck” effects

Transitional bigram probability (r = .771)



- Transitional bigram probability substantially better
- Initial/final restrictions, and rare vs. common bigrams

A recurring finding

- For attested (common vs. rare) sequences, transitional bigram probability is hard to beat as a model of acceptability judgments
  - Baseline: a simple model that sets the bar for more sophisticated models
- Important to remember: most acceptability experiments focus on acceptable or nearly-acceptable items

Positional bigram probability

Position matters

- Often, the probability of a given sequence depends on its position within the word
  - E.g., [ʒɻ] is much more common word-finally than word-initially
- A certain amount of positional dependence is already dealt with by using transitional probabilities
  - P(ɻ|ʒ) may be sort of high, but P(ʒ|#) is very low
- Another strategy in the literature: separate counts depending on position

Positional bigram probability

Jusczyk et al. (1994), p.

- “We operationally defined phonotactic probability based on two measures: (1) positional phoneme frequency (i.e., how often a given segment occurs in a position with a word) and (2) biphone frequency (i.e., the phoneme-to-phoneme cooccurrence probability)…All probabilities were computed based on log frequency-weighted values. The average summed phoneme probability was .1926 for the high-probability pattern list and .0543 for the low-probability pattern list.”

Positional bigram probability (*cont.*)

- “A high-probability phonotactic pattern also consisted of frequent segment-to-segment cooccurrence probabilities. In particular, we chose CVC phonetic patterns whose initial consonant-to-vowel cooccurrences and vowel-tofinal consonant cooccurrences had high probabilities of occurrence in the computerized database. For example, for the pattern /ɹɪs/, the probability of the cooccurrence /ɹ/ to /ɪ/ was high, as was the cooccurrence of /ɹ/ to /s/”

Vitevitch and Luce's Phonotactic Probability Calculator

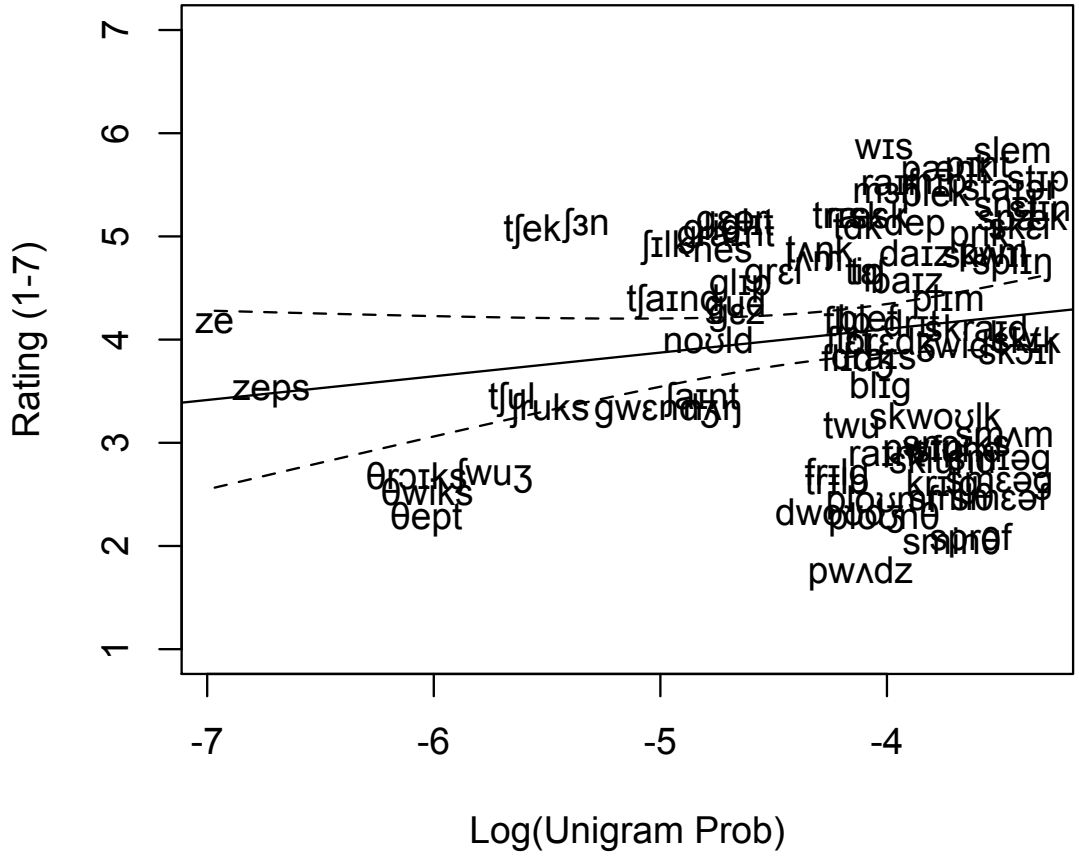
Vitevitch & Luce (2004) Phonotactic probability calculator[1]

- Position-independent bigram probability: P(xy) = $\frac{\text{(Frequency-weighted) count of } xy}{\text{(Frequency-weighted) count of all bigrams}}$
- Positional bigram probability: P(xy@position n) = $\frac{\text{(Frequency-weighted) count of } xy \text{ at position } n}{\text{(Frequency-weighted) count of all bigrams at position } n}$

- Positions: 1st, 2nd, 3rd, ... phoneme of word
  - E.g., [st] in 3rd position: *best*, *abstain*, *austere*, ...
- In point of fact, this (rather than average biphone frequency) is what Jusczyk et al. use

[1] http://www.people.ku.edu/~mvitevit/PhonoProbHome.html

Average positional bigram probability (r = .162)



- Calculations here don’t incorporate token frequency
- However: this usually makes little difference

Other concepts of position

Sensitivity to syllable structure: Pierrehumbert (1994 Labphon III): what determines probability of attestation of CCC clusters, such as [lfr] in *belfry*?

- Possibility 1: prob([lf], [fr])
- Possibility 2: argmax(prob(coda [l], onset [fr]), prob(coda [lf], onset [r]))
- Claim: metric sensitive to syllable-position works best
  - Best predictor of which CCC clusters are tested (best linear separation)
  - Also best model of data from experiment on low frequency clusters

Another syllable structure effect

A longstanding observation: onset-rhyme dissociation

- Categorical phonotactic restrictions tend to target VC, not CV combinations (Fudge 1969)

CV	/p/	/t/	/k/
[i]	peel	teal	keel
[ɪ]	pick	tick	kick
[e]	pale	tale	kale
[ɛ]	pen	ten	Ken
[æ]	pan	tan	can
[u]	pool	tool	cool
[ʊ]	put	took	cook
[o]	poke	toke	coke
[ɔ]	Paul	tall	call
[ʌ]	puff	tough	cuff
[a]	pot	tot	cot
[aɪ]	pine	tine	kine
[aʊ]	pout	tout	cow
[ɔɪ]	poise	toys	coin
[ju]	puke	—	cute

VC	/p/	/t/	/k/
[i]	leap	neat	leek
[ɪ]	lip	lit	lick
[e]	rape	rate	rake
[ɛ]	pep	pet	peck
[æ]	rap	rat	rack
[u]	coop	coot	kook
[ʊ]	—	put	book
[o]	soap	coat	soak
[ɔ]	—	taught	walk
[ʌ]	cup	cut	tuck
[a]	top	tot	lock
[aɪ]	ripe	right	like
[aʊ]	—	bout	—
[ɔɪ]	—	(a)droit	—
[ju]	—	butte	puke

Another syllable structure effect

A longstanding observation: onset-rhyme dissociation

- Categorical phonotactic restrictions tend to target VC, not CV combinations (Fudge 1969)

CV	/b/	/d/	/g/
[i]	beep	deep	geek
[ɪ]	bin	din	gill
[e]	bait	date	gait
[ɛ]	bet	deck	get
[æ]	back	Dan	gap
[u]	boon	dune	goon
[ʊ]	book	—	good
[o]	boat	dote	goat
[ɔ]	ball	doll	gall
[ʌ]	bun	done	gun
[a]	bot	dot	got
[aɪ]	buy	dine	guy
[aʊ]	bout	doubt	gout
[ɔɪ]	boy	doi(ly)	goi(ter)
[ju]	butte	—	(ar)gue

VC	/b/	/d/	/g/
[i]	grebe	lead	league
[ɪ]	bib	bid	big
[e]	babe	fade	vague
[ɛ]	Deb	bed	beg
[æ]	tab	tad	tag
[u]	tube	food	—
[ʊ]	—	could	—
[o]	robe	road	rogue
[ɔ]	daub	laud	log
[ʌ]	rub	bud	rug
[a]	cob	cod	cog
[aɪ]	bribe	ride	—
[aʊ]	—	loud	—
[ɔɪ]	—	void	—
[ju]	cube	feud	fugue

This is a grebe.

Another syllable structure effect

A longstanding observation: onset-rhyme dissociation

- Categorical phonotactic restrictions tend to target VC, not CV combinations (Fudge 1969)

CV	/m/	/n/	/ŋ/	/l/	/r/	/w/	/j/
[i]	meat	neat	—	leap	reap	weep	yeast
[ɪ]	mitt	nip	—	lip	rip	whip	yip
[e]	mate	Nate	—	late	rate	wait	yay
[ɛ]	met	net	—	let	wreck	wet	yet
[æ]	mat	nap	—	lap	rap	wax	yak
[u]	moot	newt	—	lute	route	woo	you
[ʊ]	Muslim	nook	—	look	rook	wood	you'll(?)
[o]	moat	note	—	lope	rope	woke	yoke
[ɔ]	moss	naught	—	log	Ross	walk	yawn
[ʌ]	mutt	nut	—	luck	rut	what	young
[a]	mock	knock	—	lock	rock	wand	yard
[aɪ]	mine	nine	—	line	rhyme	whine	—
[aʊ]	mouse	now	—	lout	route	wound	(yowl)
[ɔɪ]	moist	noise	—	loin	Roy	—	(yoink)
[ju]	music	—	—	—	—	—	—

Another syllable structure effect

A longstanding observation: onset-rhyme dissociation

- Categorical phonotactic restrictions tend to target VC, not CV combinations (Fudge 1969)

VC	/m/	/n/	/ŋ/	/l/	/r/	/w/	/j/
[i]	team	mean	—	teal	tear	(ewww!)	—
[ɪ]	Tim	tin	sing	till	—	—	
[e]	tame	pane	—	tale	tear	—	—
[ɛ]	hem	ten	—	tell	—	—	—
[æ]	ham	tan	tang	pal	—	—	—
[u]	tomb	tune	—	tool	tour	—	—
[ʊ]	—	—	—	full	—	—	—
[o]	tome	tone	—	toll	tore	—	—
[ɔ]	—	lawn	long	tall	—	—	—
[ʌ]	hum	ton	tongue	(skull)	—	—	—
[a]	Tom	con	—	doll ???	tar	—	—
[aɪ]	time	tine	—	tile	tire	—	—
[aʊ]	—	town	—	scowl	hour	—	—
[ɔɪ]	—	coin	—	toil	—	—	—
[ju]	fume	(im)mune	—	fuel	pure	—	—

A more systematic demonstration

Kessler and Treiman (1997)

- Statistical analysis of all CVC monosyllables in Random House dictionary
- Attempted to assess degree of over/underrepresentation of CV, VC C-C combinations
- Result in a nutshell: many VC and C-C restrictions, virtually no statistically significant CV restrictions

Making counts sensitive to syllabic position

- Pierrehumbert (1994): biphone probability across a syllable boundary may not matter
  - Or, may have less importance, at least
- Kessler and Treiman (1997): biphone probability across onset-nucleus boundary may not matter

Another way to get some syllable effects

Sequences larger than biphones

- l.f vs. lf. $=$ lf$\begin{bmatrix} +\text{son} \\ +\text{cont} \end{bmatrix}$ vs. lf[$-$son], lf[$-$cont]
- Or more generally: gradient *lf] decomposed
  - High: lfa, lfi, lfu, … lfr, lfl
  - Low: lfm, lfn, lfs, lft, …
- An immediate problem: sparse data
  - Even if […l][f…] is “perfect”, it’s not guaranteed to co-occur with all surrounding vowel contexts
  - Accidental gaps: not ruled out by any principle, just happen to be unattested given finite set of words
  - N-grams of larger length have many more possibilities $\rightarrow$ lower individual probabilities $\rightarrow$ often 0 attested occurrences

Sparse data at larger value of n

Strategies for coping with this (smoothing)

- Discounting: steal some probability mass from more frequent items, to give to unattested items
- Deleted interpolation: combine longer and shorter *n*-grams
  - $\mathrm{P}(\mathrm{c}|\mathrm{ab}) = \omega_1\mathrm{P}(\mathrm{c}|\mathrm{ab}) + \omega_2\mathrm{P}(\mathrm{c}|\mathrm{b}) + \omega_3\mathrm{P}(\mathrm{c})$
- Back-off: combine information from lower orders only if higher order is unattested
  - $\mathrm{P}(\mathrm{c}|\mathrm{ab}) =$

$$\left\{ \begin{array}{ll} \mathrm{P}(\mathrm{c}|\mathrm{ab}) & \text{if count(abc)} > 0 \\ \alpha_1\mathrm{P}(\mathrm{c}|\mathrm{b}) & \text{if count(abc)} = 0 \text{ and count(bc)} > 0 \\ \alpha_2\mathrm{P}(\mathrm{c}) & \text{otherwise} \end{array} \right\}$$

☞ See Jurafsky and Martin chapter for an overview

Non-local dependencies

Newport and Aslin (2004) Learning at a distance I.

- Constructed 3-syl stimuli: 5 fixed σ_1-σ_3 frames, variable σ_1
  - bi $\left\{\begin{array}{l}\text{di}\\ \text{ku}\\ \text{to}\\ \text{pa}\end{array}\right\}$ te, gu $\left\{\begin{array}{l}\text{di}\\ \text{ku}\\ \text{to}\\ \text{pa}\end{array}\right\}$ do, pi $\left\{\begin{array}{l}\text{di}\\ \text{ku}\\ \text{to}\\ \text{pa}\end{array}\right\}$ ra, etc.
  - Transitional prob from σ_1 to $\sigma_2 = .25$
  - Transitional prob from σ_2 to $\sigma_3 = \sigma_3$ to $\sigma_1 = .20$
  - Transitional prob. from σ_1 to $\sigma_3 = 1.00$
  - See Table 1, p. 132

Non-local dependencies (*cont.*)

- Adults exposed for 21 mins, tested on real vs. part-words
  - Adults: can just ask in randomized list: word or not?
- Result: at or below chance performance!
  - Even with more training, simpler languages, etc.
  - Only evidence of learning is ability to distinguish real words from non-words (syls in completely non-occurring orders), which can be distinguished using local transitional probabilities

Non-local dependencies

Comparison: non-local consonant or vowel dependencies

- Constructed languages with $C_1C_2C_3$ frames, variable vowels—or vice versa
  - $p\begin{Bmatrix} a \\ o \end{Bmatrix}g\begin{Bmatrix} i \\ u \end{Bmatrix}t\begin{Bmatrix} æ \\ e \end{Bmatrix}$, $d\begin{Bmatrix} a \\ o \end{Bmatrix}k\begin{Bmatrix} i \\ u \end{Bmatrix}b\begin{Bmatrix} æ \\ e \end{Bmatrix}$
  - $\begin{Bmatrix} p \\ d \end{Bmatrix}a\begin{Bmatrix} g \\ k \end{Bmatrix}u\begin{Bmatrix} t \\ b \end{Bmatrix}e$, $\begin{Bmatrix} p \\ d \end{Bmatrix}o\begin{Bmatrix} g \\ k \end{Bmatrix}i\begin{Bmatrix} t \\ b \end{Bmatrix}æ$
- Frames don't exhibit harmony or any such phonological affinity; merely a statistical reliability
- As before, test words vs. part-words
- Result: good discrimination in both conditions!

Perhaps a substantive bias to attend to certain co-occurrences, rather than others?

Neighborhood models

A different approach to predicting wordlikeness

- Counts we have considered so far are all *combinatoric*
  - Probability of cooccurrence of combinations of sounds (local or non-local) → phonotactic probability
  - Intuition: nonce words sound more plausible if they contain well-supported combinations of sounds
  - E.g., [mɪp] is very likely because many words start with [#m], many words have [mɪ], many words have [ɪp], etc.
- A different type of metric: similarity to existing words
  - Intuition: nonce words sound more plausible if they sound similar to existing words
  - E.g., [mɪp] is very wordlike because it sounds like [nɪp], [mɪt], [mæp], etc.

Neighborhood density

- Neighborhood density: The number of words that differ from target word by one change (Greenberg and Jenkins, 1964; Coltheart et al., 1977; Luce, 1986)
  - Change one segment: *plan* ∼ *clan*, *plane*, *plaque*
  - Add one segment: *plan* ∼ *plant*
  - Delete one segment: *plan* ∼ *pan*
- A crude but widely used estimate of similarity to the lexicon

Greenberg and Jenkins (1964)

Greenberg, J. & J. Jenkins (1964) Studies in the psychological correlates of the sound system of American English. *Word* 20, pp. 157-177.

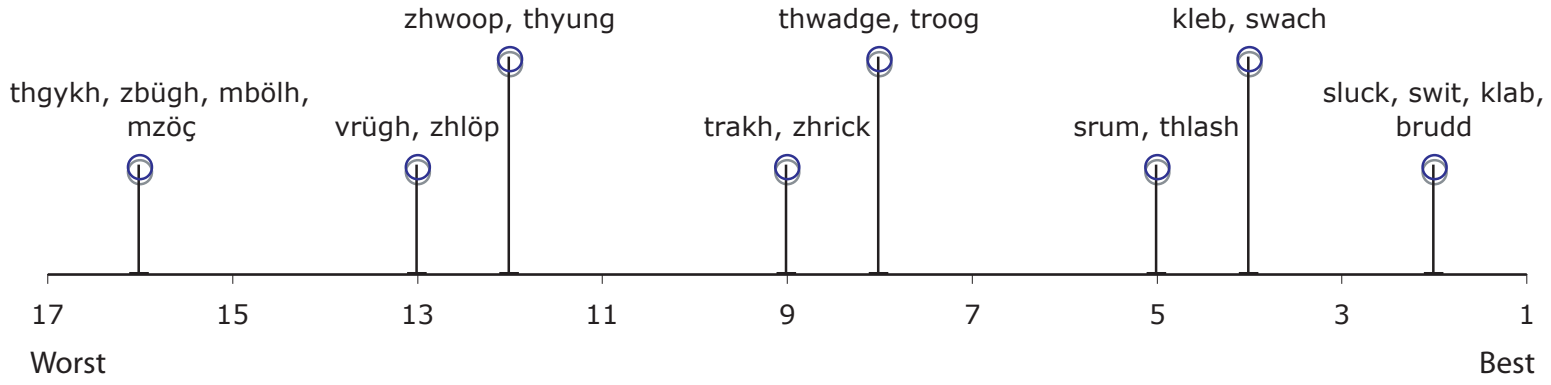
- One of the first attempts to collect systematic data about intermediate degrees of “Englishness”
- Hypothesis: novel words sound better, the closer they are to existing words

Greenberg and Jenkins (1964) (*cont.*)

- Closer = fewer differences, or fewer modifications you have to make to get to the nearest existing word
  - E.g., *clab* could be turned into *slab*, *crab*, *club*, *clam*, or *c_ab* by simply changing one sound
  - *cleb* requires two or more changes (*crab*, *clam*, *cleanse*, etc.)
- Number of similar-sounding words also makes a difference

Greenberg and Jenkins (1964) (*cont.*)

- Made up words predicted to fall along a scale of "Englishness"



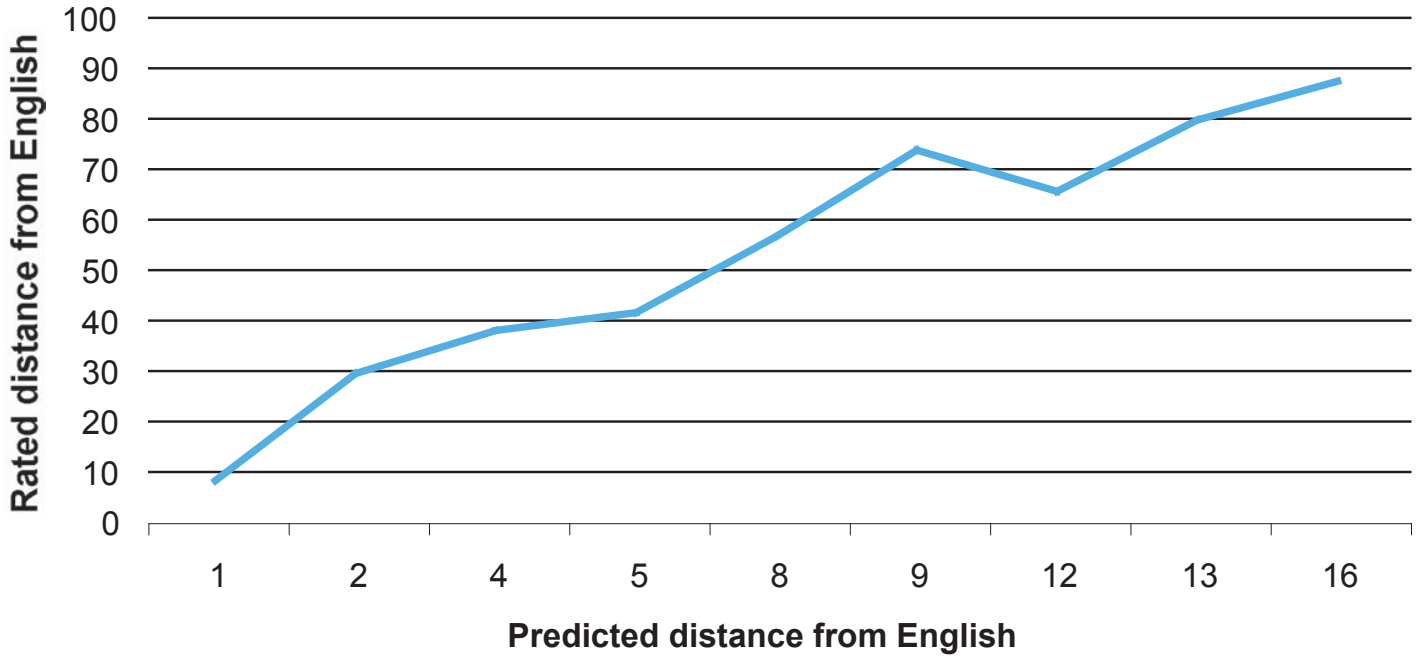
- Asked participants to rate words according to "how far they were from English" (low score = close, high score = far)

Greenberg and Jenkins (1964) (*cont.*)

- Tried various ways: scale of 0–10, arbitrary scales of the subjects' devising, etc., but always got the same results…

Greenberg and Jenkins (1964) (*cont.*)

- Results: fairly strong relationship between calculated distance and perceived distance



Greenberg and Jenkins (1964) (*cont.*)

(Ratings for distance 5 are artificially low because the made-up word *thlash* was heard by some participants as *flash*, and rated as very English-like)

Crude measure: neighborhood density

- Greenberg and Jenkins: scale reflecting how easily transformed the word is into existing words
- Alternative approach: number of single-edit neighbors (Coltheart et al. 1977; Luce 1986)
  - Neighborhood density (NNB): number of existing words that differ by one phoneme addition, deletion, or substitution
  - Perhaps weighted by frequency, etc. (or similarity: more on this below)

Neighborhood density

- Colab interlude: a script that calculates number of neighbors in CELEX
- Predictions for Albright and Hayes (2003) words

The philosophy behind neighborhood density

Neighborhood effects in lexical/memory-based models

- Claim: reactions to nonce words are not the result of learned statistical knowledge calculated over the lexicon
- Rather, they are the result of an (implicit) attempt to classify the new word
  - Is it English, or not?
  - Items that are similar to many existing words receive lots of support from the lexicon
  - Items that are not similar to any existing words receive no support
- Essentially a by-product of activation of similar words

Exemplar models

Exemplar models

- Data is stored in detailed representations that encode many aspects of the experience
- Over time, a large number of exemplars build up
- New experiences or intents activate similar existing exemplars
- These influence how the word is perceived/categorized/identified/produced

Hermann Paul: *Prinzipien der Sprachgeschichte*

"In order to understand the phenomenon which we usually designate as sound-change, we must get a clear idea of the … processes which operate in the production of groups of sound…In the first place, the movements of the organs of language…; secondly, the series of sensations by which these movements are necessarily accompanied—the 'motory sensation' (*bewegungsgefühl*); thirdly, the sensations of tone produced in the hearers…These sensations are, of course, not merely physiological processes, but psychological as well. Even after the physical excitement has passed away, these sensations leave a lasting psychical effect, viz., in the shape of memory-pictures (*erinnerungsbilder*)…[T]hese set up a connexion of cause and effect between the earlier and later production of the same combination of sounds. The memory-picture left behind by the sensation of the movement carried out before is that which renders possible the

In more modern terms

- Medin and Schaffer (1978) Context theory of classification learning

  > “…a probe stimulus functions as a retrieval cue to access information stored with stimuli similar to the probe.”

- Motivations for such an approach
  - Classification and prototypicality judgments often influenced by information that is seemingly “extraneous” information, from the point of the relevant rule (e.g., birds, prime numbers)
  - Judgments are gradient—not a binary classification

The Generalized Context Model (Nosofsky, 1986)

Background: Luce choice rule (stimulus identification)

$$\mathrm{P}(\mathrm{response}_j|\mathrm{stimulus}_i) = \frac{\mathrm{bias}_j \times \eta_{ij}}{\sum \mathrm{bias}_k \times \eta_{ik}}$$

- Probability of labeling $\mathrm{stimulus}_i$ as item j depends on relative similarity of stimulus to j vs. to other items
- $\eta_{ij} =$ perceptual similarity between i and j (possibly non-monotonic function of physical distance between i and j)
- $\eta_{ij} = \mathrm{e}^{-sensitivity \cdot distance_{ij}}$, or $\mathrm{e}^{-sensitivity \cdot distance_{ij}^2}$ (two options)
- Sensitivity $=$ a parameter (found by fitting)
- Distance: in some space (e.g., string edit distance)

The Generalized Context Model (Nosofsky, 1986)

Context model of classification (=category identification)

$$P(\text{response}_J|\text{stimulus}_i) = \frac{\text{bias}_J \times \sum_{j \in J} \eta_{ij}}{\sum_K (\text{bias}_K \times \sum_{k \in K} \eta_{ik})}$$

- Probability of classifying stimulus *i* as member of category J depends on relative similarity of *i* to members of category *J* vs. similarity to members of all other categories

Exemplar models

- Simple mechanism, easier to harness for linguistic phenomena than for others
  - Vowel identification (Johnson, and others)
  - Inflectional class (Nakisa et al., 1997; Albright and Hayes, 2003)
- In the case of gradient acceptability of novel words, however, it is not so hard to see how one might frame the problem
- Intuition: if a nonsense word has a lot of existing neighbors, it should sound better
  - The more neighbors, the better
  - The more similar those neighbors are, the better

Generalized Neighborhood Model

Bailey and Hahn (2001) Plausibility of novel word w_i

$$\propto \sum_{w_j \in lexicon} \text{weight}(w_j) \times \text{perceived sim}(w_i, w_j)$$

- As above, perceived similarity depends on physical distance
  - Perceived similarity $= e^{-\text{distance}}$ (or $e^{-\text{distance}^2}$, not considered)
- Denominator can be ignored here
  - What would it refer to?
- Weights of words related to frequency (non-linearly?)

Generalized Neighborhood Model (*cont.*)

- Bailey and Hahn propose quadratic fit: $\alpha \times \text{freq}^2 + \beta\text{freq} + \gamma$
- Allows for parabolic relations (monotonic or exponential increase/decrease, or greater/lesser influence of mid-range)

Generalized Neighborhood Model

$$\text{Score of } w_i = \sum_{w_j \in lexicon} \text{weight}(w_j) \times \text{perceived sim}(w_i, w_j)$$

$$= \sum_{w_j} \text{weight}(w_j) \times e^{-sensitivity \cdot d_{i,j}}$$

$$= \sum_{w_j} (\alpha f_j^2 + \beta f_j + \gamma) \times e^{-sensitivity \cdot d_{i,j}}$$

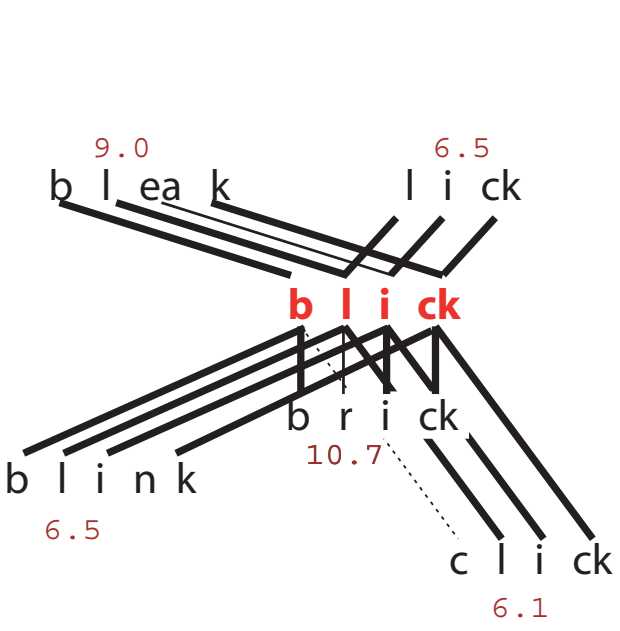
(see Bailey and Hahn, 2001, p. 572)

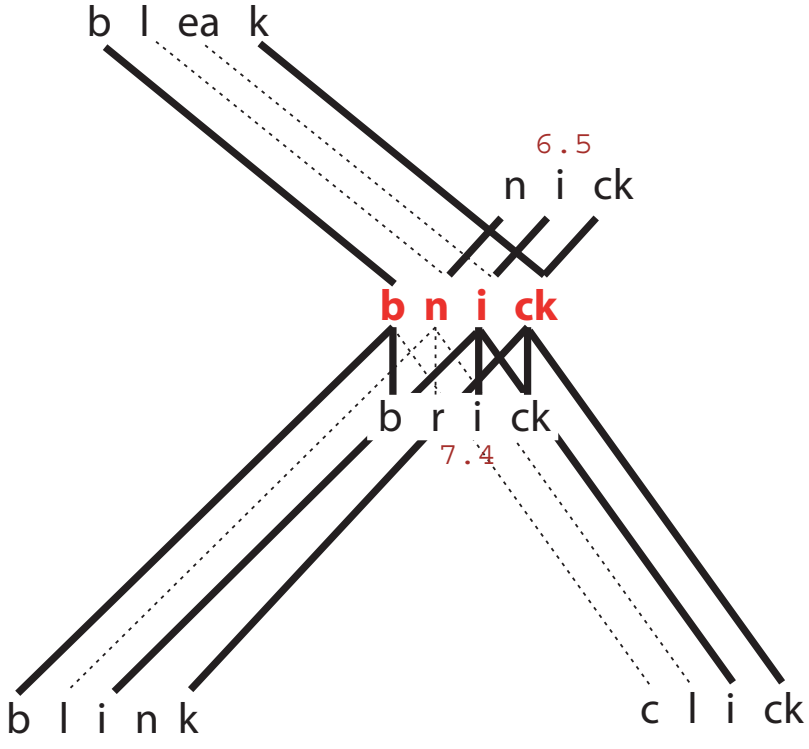
Generalized Neighborhood Model

How do we measure the perceived similarity of two words?

- Greenberg and Jenkins (and many subsequent researchers) define neighbors simply in terms of number of substitutions
  - Novel *clab*: *cab*, *crab*, *club*, *class* all equally close
- This seems too crude; not all substitutions have equal consequences for similarity of the resulting pairs
- We need a better way to calculate the similarity of two words

Similarity as overlap





- Mismatches depend on relative similarity of segments
  - ☞ E.g., *blick* ~ *brick* more similar than *blick* ~ *click*, because *l*~*r* more similar than *b*~*k*
- *Bnick* has less overlap with existing words, and mismatches are more serious; gets less support

Calculating similarity

- Goodness of matches between segments is assessed using natural-class based similarity metric of Frisch, Broe, and Pierrehumbert (2004)
- Degree of overlap between whole words is assessed by finding minimum string edit distance

Substitution costs based on similarity

Evidence that humans care about something like featural distance

- Greenberg and Jenkins (1964)
  - How similar are *ba* and *da*? *ba* and *pa*? *ba* and *ta*?
  - Results:
    - *b*∼*d*∼*g* rated very similar to one another
    - *b*∼*p*, *d*∼*t*, *g*∼*k* more similar than *b*∼*t*, *k*∼*p*, etc.
  - Conclusion: one feature change more similar than two feature changes

Substitution costs based on similarity (*cont.*)

- Hahn and Bailey (2005) What makes words sound similar
  - Choose more similar pair: XA (1 change) or XB (2 changes)
    - Onsets: *plimp*∼*flimp* or *plimp*∼*slimp*
    - Codas: *plip*∼*plib* or *plip*∼*plig*
  - Subjects generally go for the one-change pairs

Tversky (1977) Feature contrast model

Distance = shared − unshared features:

$$\mathrm{d}_{a,b} = \theta \cdot f(a \cap b) - \alpha \cdot f(a - b) - \beta \cdot f(b - a)$$

- $a \cap b$ = set of features shared by a and b
- $a - b$ = set of features of a but not b
- $f(a)$ = weighted function of feature contributions
- Parameters θ, α, β depend on task (more important to share at least some attributes? have no differences? etc.)
- Closely related to ratio mode (a la Frisch et al):

$$\mathrm{d}_{a,b} = \frac{f(a \cap b)}{f(a \cap b) + \alpha \cdot f(a - b) + \beta \cdot f(b - a)}$$

The task

- The task:
  - Find a set of feature weights *f*, and parameters θ, α, β, such that sim(*x*,*y*) values for all pairs *x*,*y* match observed values as closely as possible
  - Observed similarities: perceptual confusability, interaction in speech errors, avoidance in root restrictions, judged similarity, etc.
- Challenges
  - Many free parameters (need good search procedure)
  - What is the correct set of features?

Natural class-based similarity

Similarity of pairs of sounds: metric based on natural classes (Frisch et al., 2004)

- Try all possible combinations of feature values
  - All subsets of features, all combinations of values
- See which combinations result in distinct natural classes
  - Collect set of distinct classes
- Sounds are similar if they are grouped together in many natural classes
  - *m* and *n* are both voiced, both nasal, both sonorant, both stops, etc.

$$\text{Similarity} = \frac{\text{shared natural classes}}{\text{shared} + \text{unshared natural classes}}$$

Colab interlude

- Calculate the similarity of vowels in a Spanish-like 5 vowel system using natural-class based similarity
  - i, e, a, o, u
  - Features: [±high], [±low], [±back]
- Verify that the addition of [±round] does not affect that calculation
- Script to calculate similarity: SimilarityCalculator.pl

Natural classes are not enough

A possibly telling datum from Hahn and Bailey (2005)

- 75% preference for [lɪl]∼[vɪl] (manner + place change) over [lɪl]∼[zɪl] (manner and stridency change)
- Not predicted by Frisch et al model:

Comparison	Shared	Unshared	Similarity
l∼v	5	31	0.161
l∼z	11	32	0.344

- Suggests that stridency matters more than place
  - ☞ See also Coon and Gallagher (2007)

String edit distance

Now that we know the similarity of pairs of segments, we need to figure out how to line up words so that the most similarity segments are in correspondence with one another

- Basic idea: alignment can be calculated by figuring out the smallest number of changes needed to change one string to another
- If two strings share material, don’t need to change it
- Unshared material must be deleted, inserted, or substituted

String alignment

Alignments = transformations: *spling* vs. *slink*

s	*p*	*l*	*ɪ*	*ŋ*	—
s	—	*l*	*ɪ*	*ŋ*	*k*

s	*p*	*l*	*ɪ*	*ŋ*	
|		|	|	|	
s		*l*	*ɪ*	*ŋ*	*k*

1. Leave *s* unchanged
2. Delete *p*
3. Leave *l* unchanged
4. Leave *ɪ* unchanged
5. Leave *ŋ* unchanged
6. Insert *k*

- Aligned segments = substituted or unchanged
- Unaligned segments = inserted or deleted
- Goal: use segmental similarity to decide what should be aligned with what

String alignment

The task:

- Analyze correspondence as a sequence of substitutions, insertions, and deletions
- In practice, we usually want the *shortest* sequence of alignments/changes
- That is, the *optimal* alignment
- We'll start first by ignoring phonetic distance, and then incorporate it at the end

String alignment

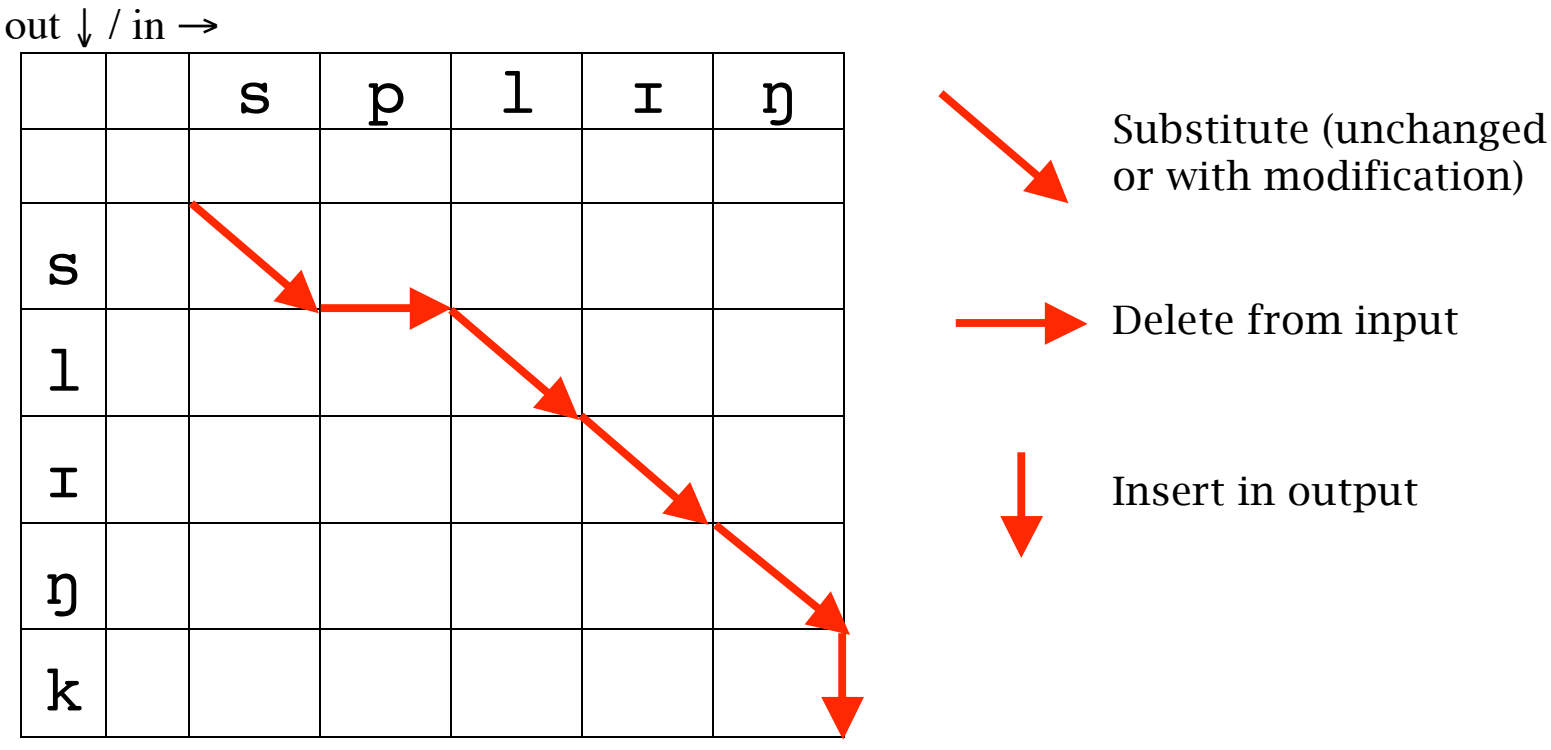
Chart to calculate alignment

out ↓ / in →

		s	p	l	ɪ	ŋ
s						
l						
ɪ						
ŋ						
k						

String alignment

Optimal path



String alignment

Calculating substitution and insertion/deletion costs

out ↓ / in →

		s	p	l	ɪ	ŋ
	0	0.5	1.0	1.5	2.0	2.5
s	0.5					
l	1.0					
ɪ	1.5					
ŋ	2.0					
k	2.5					

Substitute (unchanged or with modification)

Delete from input

Insert in output

subst cost	del cost
insert cost	

String alignment

Calculating substitution and insertion/deletion costs

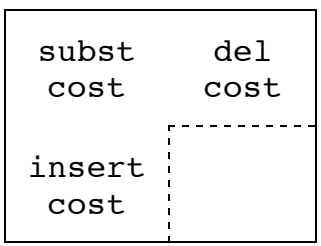
out ↓ / in →

		s	p	l	ɪ	ŋ
	0	0.5	1.0	1.5	2.0	2.5
s	0.5	0 .5 .5				
l	1.0					
ɪ	1.5					
ŋ	2.0					
k	2.5					

Substitute (unchanged or with modification)

Delete from input

Insert in output



String alignment

Calculating substitution and insertion/deletion costs

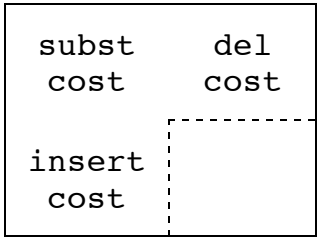
out ↓ / in →

		s	p	l	ɪ	ŋ
	0	0.5	1.0	1.5	2.0	2.5
s	0.5	**0** **.5** / **.5**	**1** **.5** / **.5**			
l	1.0					
ɪ	1.5					
ŋ	2.0					
k	2.5					

Substitute (unchanged or with modification)

Delete from input

Insert in output



String alignment

Calculating substitution and insertion/deletion costs

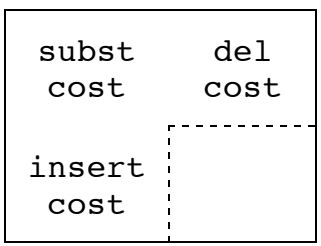
out ↓ / in →

		s	p	l	ɪ	ŋ
	0	0.5	1.0	1.5	2.0	2.5
s	0.5	**0 .5** / **.5**	**1 .5** / **.5**	**1 .5** / **.5**	**1 .5** / **.5**	**1 .5** / **.5**
l	1.0					
ɪ	1.5					
ŋ	2.0					
k	2.5					

Substitute (unchanged or with modification)

Delete from input

Insert in output



String alignment

Calculating substitution and insertion/deletion costs

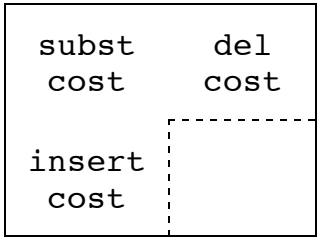
out ↓ / in →

		s	p	l	ɪ	ŋ
	0	0.5	1.0	1.5	2.0	2.5
s	0.5	**0** **.5** **.5**	**1** **.5** **.5**	**1** **.5** **.5**	**1** **.5** **.5**	**1** **.5** **.5**
l	1.0	**1** **.5** **.5**	**1** **.5** **.5**	**0** **.5** **.5**	**1** **.5** **.5**	**1** **.5** **.5**
ɪ	1.5					
ŋ	2.0					
k	2.5					

Substitute (unchanged or with modification)

Delete from input

Insert in output



String alignment

Calculating substitution and insertion/deletion costs

out ↓ / in →

		s	p	l	ɪ	ŋ
	0	0.5	1.0	1.5	2.0	2.5
s	0.5	**0** **.5** / **.5**	**1** **.5** / **.5**	**1** **.5** / **.5**	**1** **.5** / **.5**	**1** **.5** / **.5**
l	1.0	**1** **.5** / **.5**	**1** **.5** / **.5**	**0** **.5** / **.5**	**1** **.5** / **.5**	**1** **.5** / **.5**
ɪ	1.5	**1** **.5** / **.5**	**1** **.5** / **.5**	**1** **.5** / **.5**	**0** **.5** / **.5**	**1** **.5** / **.5**
ŋ	2.0	**1** **.5** / **.5**	**1** **.5** / **.5**	**1** **.5** / **.5**	**1** **.5** / **.5**	**0** **.5** / **.5**
k	2.5	**1** **.5** / **.5**	**1** **.5** / **.5**	**1** **.5** / **.5**	**1** **.5** / **.5**	**1** **.5** / **.5**

Substitute (unchanged or with modification)

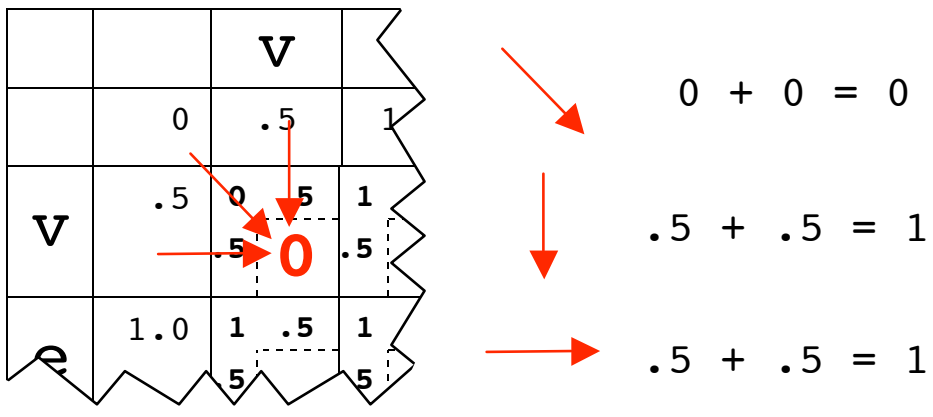
Delete from input

Insert in output

subst cost	del cost
insert cost	

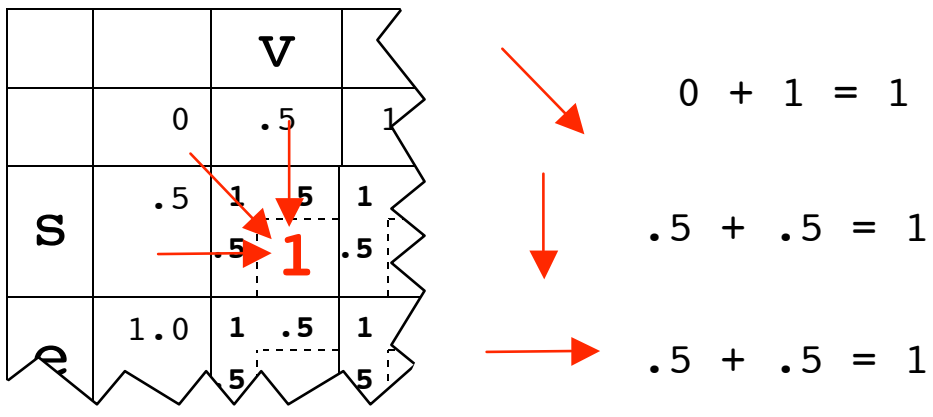
String alignment

Center value = minimum of three corners



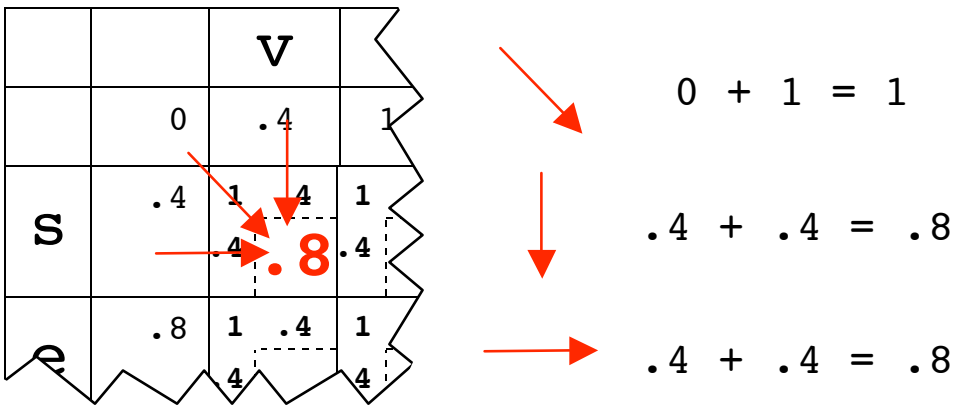
String alignment

Center value = minimum of three corners



String alignment

Center value = minimum of three corners



String alignment

Center value = minimum of three corners

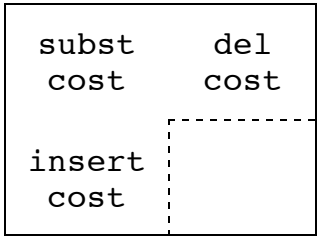
out ↓ / in →

		s	p	l	ɪ	ŋ
	0	0.5	1.0	1.5	2.0	2.5
s	0.5	0 .5 / .5 0	1 .5 / .5	1 .5 / .5	1 .5 / .5	1 .5 / .5
l	1.0	1 .5 / .5	1 .5 / .5	0 .5 / .5	1 .5 / .5	1 .5 / .5
ɪ	1.5	1 .5 / .5	1 .5 / .5	1 .5 / .5	0 .5 / .5	1 .5 / .5
ŋ	2.0	1 .5 / .5	1 .5 / .5	1 .5 / .5	1 .5 / .5	0 .5 / .5
k	2.5	1 .5 / .5	1 .5 / .5	1 .5 / .5	1 .5 / .5	1 .5 / .5

Substitute (unchanged or with modification)

Delete from input

Insert in output



String alignment

Center value = minimum of three corners

out ↓ / in →

		s	p	l	ɪ	ŋ
	0	0.5	1.0	1.5	2.0	2.5
s	0.5	0 .5 / .5 0	1 .5 / .5 .5	1 .5 / .5	1 .5 / .5	1 .5 / .5
l	1.0	1 .5 / .5	1 .5 / .5	0 .5 / .5	1 .5 / .5	1 .5 / .5
ɪ	1.5	1 .5 / .5	1 .5 / .5	1 .5 / .5	0 .5 / .5	1 .5 / .5
ŋ	2.0	1 .5 / .5	1 .5 / .5	1 .5 / .5	1 .5 / .5	0 .5 / .5
k	2.5	1 .5 / .5	1 .5 / .5	1 .5 / .5	1 .5 / .5	1 .5 / .5

Substitute (unchanged or with modification)

Delete from input

Insert in output

subst cost	del cost
insert cost	

String alignment

Center value = minimum of three corners

out ↓ / in →

		s	p	l	ɪ	ŋ
	0	0.5	1.0	1.5	2.0	2.5
s	0.5	0 .5 .5 0	1 .5 .5 .5	1 .5 .5 1.0	1 .5 .5 1.5	1 .5 .5 2.0
l	1.0	1 .5 .5	1 .5 .5	0 .5 .5	1 .5 .5	1 .5 .5
ɪ	1.5	1 .5 .5	1 .5 .5	1 .5 .5	0 .5 .5	1 .5 .5
ŋ	2.0	1 .5 .5	1 .5 .5	1 .5 .5	1 .5 .5	0 .5 .5
k	2.5	1 .5 .5	1 .5 .5	1 .5 .5	1 .5 .5	1 .5 .5

Substitute (unchanged or with modification)

Delete from input

Insert in output

subst cost	del cost
insert cost	

String alignment

Center value = minimum of three corners

out ↓ / in →

		s	p	l	ɪ	ŋ
	0	0.5	1.0	1.5	2.0	2.5
s	0.5	0 .5 .5 0	1 .5 .5 .5	1 .5 .5 1.0	1 .5 .5 1.5	1 .5 .5 2.0
l	1.0	1 .5 .5 .5	1 .5 .5 1.0	0 .5 .5 .5	1 .5 .5 1.0	1 .5 .5 1.5
ɪ	1.5	1 .5 .5 1.0	1 .5 .5 1.5	1 .5 .5 .5	0 .5 .5 .5	1 .5 .5 1.5
ŋ	2.0	1 .5 .5 1.5	1 .5 .5 2.0	1 .5 .5 1.5	1 .5 .5 1.0	0 .5 .5 .5
k	2.5	1 .5 .5 2.0	1 .5 .5 2.5	1 .5 .5 2.0	1 .5 .5 1.5	1 .5 .5 1.0

Substitute (unchanged or with modification)

Delete from input

Insert in output

subst cost	del cost
insert cost	

String alignment

Paths with smallest costs

out ↓ / in →

		s	p	l	ɪ	ŋ
	0	0.5	1.0	1.5	2.0	2.5
s	0.5	0 / 0	.5 / .5			
l	1.0			0 / .5		
ɪ	1.5				0 / .5	
ŋ	2.0					0 / .5
k	2.5					.5 / 1.0

Substitute (unchanged or with modification)

Delete from input

Insert in output

subst cost	del cost
insert cost	

String alignment

Finally: phonetically sensible substitution costs

- Substitution cost(x,y) $= 1 -$ similarity(x,y)
- Choice of aligning vs. inserting/deleting determined by relative cost of indel—e.g., hypothetically with indel$=$.5:

Pair	Similarity	Sub cost	Alignment
b, *p*	.8	.2	Sub *b*$\rightarrow$*p*
b, *v*	.6	.4	Sub *b*$\rightarrow$*v*
b, *m*	.5	.5	Sub b$\rightarrow$m, or delete *b*/insert *m*
b, *f*	.3	.6	Delete *b*/insert *f*

String alignment

Sample of sensible substitution costs

out ↓ / in →

		s	p	l	ɹ	ŋ
	0	0.5	1.0	1.5	2.0	2.5
s	0.5	0 .5 / .5 0	.86 .5 / .5 .5	.84 .5 / .5 1.0	.98 .5 / .5 1.5	.87 .5 / .5 2.0
l	1.0	.84 .5 / .5 .5	.93 .5 / .5 .9	0 .5 / .5 .5	.89 .5 / .5 1.0	.68 .5 / .5 1.5
ɹ	1.5	.98 .5 / .5 1.0	.97 .5 / .5 1.4	.89 .5 / .5 .5	0 .5 / .5 .5	.90 .5 / .5 1.5
ŋ	2.0	.87 .5 / .5 1.5	.83 .5 / .5 1.8	.68 .5 / .5 1.5	.89 .5 / .5 1.0	0 .5 / .5 .5
k	2.5	.84 .5 / .5 2.0	.44 .5 / .5 1.9	.92 .5 / .5 2.0	.97 .5 / .5 1.5	.77 .5 / .5 1.0

Substitute (unchanged or with modification)

Delete from input

Insert in output

subst cost	del cost
insert cost	

(Doesn't change anything in this case)

Colab interlude

Running and testing an implementation of the GNM

- Calculate similarity values for English inventory
- See alignment in action
- Use GNM to derive predictions for Albright and Hayes (2003) words

Testing the GNM

Bailey & Hahn (2001)

- Motivating suspicion: many purported effects of phonotactic probability may actually be neighborhood effects
  - In point of fact, NNB and n-gram probability are often highly correlated (why?)
  - Most studies test for effects of one or the other, but don’t directly compare the two
  - Previous tests for neighborhood effects hampered by crude definition of neighborhoods
- Strategy
  - Collect ratings for a huge bunch of nonce words
  - Compare predictive power of GNM and n-gram models

Nonce word experiment

Bailey & Hahn (2001)

- Made up 22 *isolates*: 2 edits from any existing word
- Added 250 *near-misses*: modified isolates to create words that had at least one existing neighbor
  - E.g., [dɹɒlf] → [dɒlf], [drɪlf], [dɹɒf], etc.
- Collected "wordlikeness" judgments
  - "How typical-sounding is ___?"
  - One orthographic task, one auditory task

Strategy for teasing apart neighbors from sequences

Multiple regression modeling

- Start by considering correlation of many different potential predictors—e.g.,
  - Phonological phone, bigram, trigram transitional probability
  - Orthographic letter, bigram, trigram transitional probability
  - Onset-rhyme transitional probability
  - Number of neighbors (traditional metric)
  - GNM score
- Then considered factors in combination, checking for ability of successive factors to improve on simpler models

Bailey & Hahn results

Individual factors

- Log joint transitional probability (bigram): $r^2 = .19$ ($r \approx .44$)
- GNM w/distances based on string alignment: $r^2 = .22$ ($r \approx .47$)
  - Slightly better, though care needed since model produces many nearly identical values
- Other metrics not nearly as good (see Table 2, p. 577)

Combining factors

- Sequence probabilities alone: $r^2 = .23$
- Adding traditional NNB: no improvement
- Adding GNM scores: $r^2 = .38$
  - Non-monotonic frequency effect: mid frequencies contribute most
  - When token frequency removed, $r^2 = .36$
- Substantially overlapping predictions

Unique contribution of phonotactics	.09
Unique contribution of GNM	.15
Overlapping predictions	.14

What do we conclude from all this?

- Significant effects of both neighbor activation and also knowledge of probability of sequences (independent of particular words)
  - Contrary to Bailey and Hahn's expectations, neither reducible to the other
- Bailey and Hahn: majority of effect is neighbors, while contribution of phonotactics is relatively smaller
  - Assessment depends on how we credit ambiguous overlapping portion of variance, however
- Role of token frequency also claimed to be significant
  - Effect is extremely small, however, and non-monotonic
  - Response curve is unlike how activation works in lexical access
  - Not clear that this supports the the underlying premise of the exemplar model

Why neighborhoods are unlikely to be sufficient

Why we might have predicted this result from the start

- Certainly, lexical access is a real effect
  - May depend on task, but no reason to preclude possibility of lexical neighbor effects
  - So, not too surprising if we see a neighborhood effect
- However, not all acceptability judgments can be reduced to number of neighbors
  - E.g., [fruː] vs. [sruː]
  - [sruː] has more neighbors, because more licit *sC*-clusters
  - Yet it also contains a phonotactic violation: **sr*

Why neighborhoods are unlikely to be sufficient (*cont.*)

- Undeniable role for sequence probability (whether learned statistically or based on markedness preferences)

Why neighborhoods are unlikely to be sufficient (*cont.*)

- Bailey and Hahn don’t actually test illegal sequences
  - And few studies bother to include *truly* unacceptable words, where sequential models can guarantee consistently low scores

Possible refinements

- Variable indel
  - Segment identity: insertion cost(ə) $<$ insertion cost(*a*)
  - Context: insertion cost(ə)/T___ R $<$ insertion cost(ə)/S___ T
  - Medial vs. peripheral indel (contiguity)
- Incorporating positional/prosodic structure

Relative cost of insertion and deletion

Gentner and Markman (1995)

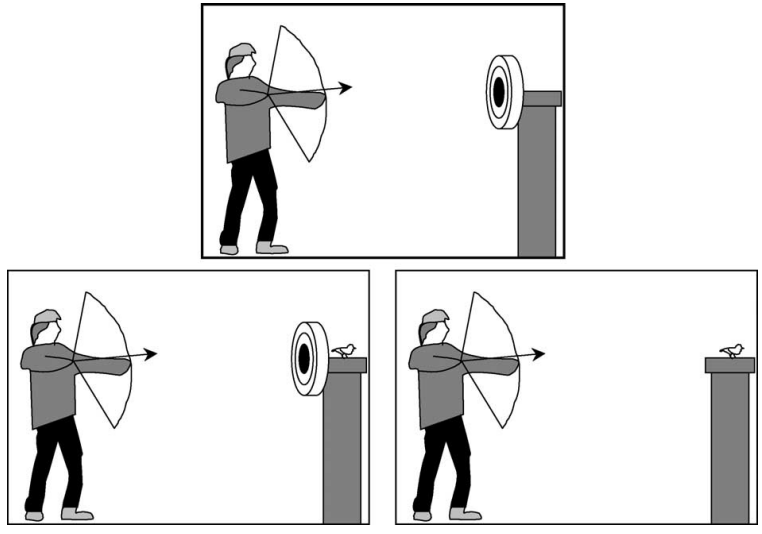


Fig. 5. Insertion and substitution in a scene depicting predicate-argument relations between objects (after Markman & Gentner, 1996, Fig. 4, p. 242).

- Subjects overwhelmingly (88%) prefer inserting bird (left) over swapping target out for bird (right)
- Claim: judgment driven by similarity of aligned elements; little penalty for additional

Relative cost of insertion and deletion

Hahn and Bailey (2005): tested phonological equivalent

- Compared insertion ([fɑːʒ]∼[flɑːʒ], [zɪtʃ]∼[zɪntʃ]) with replacement ([fɑːʒ]∼[lɑːʒ], [zɪtʃ]∼[zɪn])
- Onsets: no preference
- Codas: insertions judged more similar

What this suggests for the model

- Alignments should have relatively low indel cost
  - [sklæm] should get lots of support from *clam*, less from *scram*
- But not *too* small?
  - Bailey & Hahn get best value from GNM with indel $>$ .6

The role of structure

Another robust finding: the importance of position (Nelson and Nelson 1970, Bailey and Hahn 2005)

- Already seen above: sim([fɑːʒ],[flɑːʒ]) = sim([fɑːʒ],[lɑːʒ]), but sim([zɪtʃ], [zɪntʃ]) > sim([zɪtʃ],[zɪn])
- More directly: sim([ʃæʃ], [fæʃ]) somewhat greater than sim([ʃæʃ], [ʃæf])
- Two lines of attack
  - Enhanced perceptibility in coda position enhances differences—VERY UNLIKELY!!! (contradicts results of perceptual studies, and typological facts)
  - Greater psychological weight to VC than CV (‘rhyming’)

Another interesting effect of structure: contrast

Goldstone et al. (1991) Relational similarity and non-independence of features in similarity judgments

- Which is more like △ △ ?

A ○ ○

B □ □

Another interesting effect of structure: contrast

Goldstone et al. (1991) Relational similarity and non-independence of features in similarity judgments

- Which is more like △ △ ?

  A ○ ○

  B □ □

- Which is more like △ △ □?

  A ○ ○ □

  B □ □ □

(What kinds of predictions does this make for comparisons of words?)

Taking stock

Two approaches to modeling phonotactic acceptability

- N-grams: decompose string into subparts
- Exemplar models: align string to stored examples
- Neither one looks all that much like phonological grammars that you may be familiar with!
  - No phonological features, syllable structure, markedness constraints
- Baseline models: help show where further assumptions improve match to humans

Finding likely areas for improvement

- Weaknesses of N-gram models?

Finding likely areas for improvement

- Weaknesses of N-gram models?
  - Local vs. non-local dependencies
  - Differentiating unattested sequences: *bnick* [bnɪk] vs. *bzick* [bzɪk] vs. *nbick* [nbɪk] (all zero)

Finding likely areas for improvement

- Weaknesses of N-gram models?
  - Local vs. non-local dependencies
  - Differentiating unattested sequences: *bnick* [bnɪk] vs. *bzick* [bzɪk] vs. *nbick* [nbɪk] (all zero)
- Weaknesses of GNM?

Finding likely areas for improvement

- Weaknesses of N-gram models?
  - Local vs. non-local dependencies
  - Differentiating unattested sequences: *bnick* [bnɪk] vs. *bzick* [bzɪk] vs. *nbick* [nbɪk] (all zero)
- Weaknesses of GNM?
  - Illegal strings may align well with many existing words
  - Differentiating unattested sequences: *bnick* [bnɪk] vs. *bzick* [bzɪk] vs. *nbick* [nbɪk] (depend on existing words)

References

Albright, Adam and Bruce Hayes. 2003. Rules vs. analogy in English past tenses: A computational/experimental study. Cognition 90, 119–161.

Bailey, Todd M. and Ulrike Hahn. 2001. Determinants of wordlikeness: Phonotactics or lexical neighborhoods? Journal of Memory and Language 44, 568–591.

Chomsky, Noam and Morris Halle. 1965. Some controversial questions in phonological theory. Journal of Linguistics 1, 97–138.

References (*cont.*)

Coltheart, Max; Eileen Davelaar; Jon Torfi Jonasson; and Derek Besner. 1977. Access to the internal lexicon. In Attention and performance, ed. by Stan Dornic, volume 6, 535–555. Hillsdale, NJ: Erlbaum.

Frisch, Stefan A.; Janet B. Pierrehumbert; and Michael B. Broe. 2004. Similarity avoidance and the OCP. Natural Language & Linguistic Theory 22(1), 179–228. URL http://www.springerlink.com/openurl.asp?genre=article&id=doi:10.1023/B:NALA.0000005557.78535.3c.

Gentner, D. and A. B. Markman. 1995. Similarity is like analogy: Structural alignment in comparison. In Similarity in language, thought and perception, ed. by C. Cacciari, 111–147. Brussels: BREPOLS.

References (*cont.*)

Greenberg, Joseph H. and James J. Jenkins. 1964. Studies in the psychological correlates of the sound system of American English. Word 20, 157–177.

Hahn, Ulrike and Todd M. Bailey. 2005. What makes words sound similar? Cognition 97, 227–267.

Halle, Morris. 1978. Knowledge unlearned and untaught: What speakers know about the sounds of their language. In Linguistic Theory and Psychological Reality., ed. by Morris Halle; Joan Bresnan; and George Miller, 294–303. MIT Press.

Hockett, Charles F. 1955. A Manual of Phonology. Baltimore: Waverly Press.

References (*cont.*)

Jusczyk, Peter W.; Angela Friderici; Jeanine M.I. Wessels; Vigdis Y. Svenkerud; and Ann Marie Jusczyk. 1993. Infants' sensitivity to the sound patterns of native language words. Journal of Memory and Language 32, 402–420.

Jusczyk, Peter W.; Paul A. Luce; and Jan Charles-Luce. 1994. Infants' sensitivity to phonotactic patterns in the native language. Journal of Memory and Language 33, 630–645.

Kessler, Brett and Rebecca Treiman. 1997. Syllable structure and the distribution of phonemes in English syllables. Journal of Memory and Language 37, 295–311.

Luce, Paul A. 1986. Neighborhoods of words in the mental lexicon. Technical report, Speech Research Laboratory, Department of Psychology, Indiana University.

References (*cont.*)

Nakisa, Ramin Charles; Kim Plunkett; and Ulrike Hahn. 1997. A Cross-Linguistic Comparison of Single and Dual-Route Models of Inflectional Morphology. In Cognitive Models of Language Acquisition, ed. by P. Broeder and J. Murre. Cambridge, MA: MIT Press.

Nosofsky, Robert M. 1986. Attention, similarity, and the identification-categorization relationship. Journal of Experimental Psychology: General 115, 39–57. http://www.cogs.indiana.edu/nosofsky/pubs/1986_rmn_jep-g_attention.pdf.

Paul, Hermann. 1920. Prinzipien der Sprachgeschichte. Halle: Niemeyer, 5th edition.

References (*cont.*)

Pierrehumbert, Janet. 1994. Syllable structure and word structure. In Papers in laboratory phonology III: Phonological structure and phonetic form, ed. by P. Keating, 168–190. Cambridge: Cambridge University Press.

Vitevitch, Michael S. and Paul A. Luce. 2004. A web-based interface to calculate phonotactic probability for words and nonwords in english. Behavior Research Methods, Instruments, & Computers 36, 481–487.

Transitional probability

Saffran, Aslin, and Newport (1996) Statistical Learning by 8-Month-Old Infants. *Science* 274, pp. 1926-1928.

- Infants trained on text of 3-syllable “words”
  - E.g., *pabiku*, *tibudo*, *golatu*, and *daropi*

Transitional probability

Saffran, Aslin, and Newport (1996) Statistical Learning by 8-Month-Old Infants. *Science* 274, pp. 1926-1928.

- Infants trained on text of 3-syllable “words”
  - E.g., *pabiku*, *tibudo*, *golatu*, and *daropi*

Example

pabikugolatupabikudaropitibudogolatudaropipabikutibudopabiku

ropigolatupabikugolatutibudodaropitibudo…

Transitional probability

Saffran, Aslin, and Newport (1996) Statistical Learning by 8-Month-Old Infants. *Science* 274, pp. 1926-1928.

- Infants trained on text of 3-syllable "words"
  - E.g., *pabiku*, *tibudo*, *golatu*, and *daropi*

Example

pabikugolatupabikudaropitibudogolatudaropipabikutibudopabiku
ropigolatupabikugolatutibudodaropitibudo…

- Items controlled so no syllable was used in more than one word
- Non-final syllables always followed by the same syllable (*pa* can only come before *bi*, etc.)
- Final syllables could be followed by any other word (*ku* could be followed by *ti*, *go*, or *da*)

Transitional probability

Saffran, Aslin, and Newport (1996) Statistical Learning by 8-Month-Old Infants. *Science* 274, pp. 1926-1928.

- Infants trained on text of 3-syllable “words”
  - E.g., *pabiku*, *tibudo*, *golatu*, and *daropi*

Example

pabikugolatupabikudaropitibudogolatudaropipabikutibudopabiku
ropigolatupabikugolatutibudodaropitibudo…

- Items controlled so no syllable was used in more than one word
- Non-final syllables always followed by the same syllable (*pa* can only come before *bi*, etc.)
- Final syllables could be followed by any other word (*ku* could be followed by *ti*, *go*, or *da*)

Transitional probability

Saffran, Aslin, and Newport (1996) Statistical Learning by 8-Month-Old Infants. *Science* 274, pp. 1926-1928.

- Infants trained on text of 3-syllable “words”
  - E.g., *pabiku*, *tibudo*, *golatu*, and *daropi*

Example

pabikugolatupabikudaropitibudogolatudaropipabikutibudopabiku
ropigolatupabikugolatutibudodaropitibudo…

- Items controlled so no syllable was used in more than one word
- Non-final syllables always followed by the same syllable (*pa* can only come before *bi*, etc.)
- Final syllables could be followed by any other word (*ku* could be followed by *ti*, *go*, or *da*)

Transitional bigram probability

Example

Words: *pabiku*, *tibudo*, *golatu*, and *daropi*

pabikugolatupabikudaropitibudogolatudaropipabikutibudopabiku

ropigolatupabikugolatutibudodaropitibudo…

- Results: after two minutes of exposure, infants can reliably distinguish “words” from “part-words”
  - “Words”: strings of syls that always occur together, like *pabiku*
  - “Part-words”: strings of syls that occur together, but only occasionally (e.g., *kudaro*)
  - Distinguish: prefer to look longer at a speaker playing part-words
- Claim: in order to do this, they must be able to track (somehow) the sequencing of syllable combinations like *pa* and *bi*

Evidence that humans care about transitional probability

Aslin, Saffran & Newport (1998) Computation of conditional probability statistics by 8-month-old infants. *Psych Sci* 9, pp. 321–324.

- Same set-up as before: four words (*pabiku*, *tibudo*, *golatu*, and *daropi*), no repeated syllables
- Small change: two words occurred twice as often as the other two in the training
  - How does this change the syllable bigram probabilities? how about the transitional probabilities?
- Test: tested "part-words" vs. low frequency "words"
- Result: infants still distinguished between the two
  - As before, preferred to look longer at part-words

☞ What statistical differences might they (in principle) have been responding to?