Productivity in Analogical Change

Constantine Lignos

Computer and Information Science
Institute for Research in Cognitive Science
University of Pennsylvania

Manchester and Salford New Researchers
Forum in Linguistics, 11/3/2012

Introduction: A case for modeling

Why model language change?

- Change happens

  Unfortunately, or luckily, no language is tyrannically consistent. All grammars leak. (Sapir, 1921)

- Modeling forces us to build a mechanistic explanation of *how*
- Advances in child data availability and acquisition modeling allow for explicit models

Analogical change

- Analogical change: learning process causes a generalization to spread from one set of forms to others
- Generalization's domain narrows (Kiparsky, 1993):
  1. Applies to phrase
  2. Applies to phrase and word
  3. Applies to stem, phrase, and word
- Change under discussion: postnasal plosive deletion
  - *b/g* → ø / N_]$_{\sigma}$ (N = homorganic nasal) (Borowsky, 1993)
  - Ex.: sing: *sɪŋ *sɪŋg,* sing-er: *sɪŋɚ *sɪŋgɚ,* finger: *fɪŋgɚ *fɪŋɚ*
  - (Tricky cases in comparatives that retain /*g*/: younger, longer, stronger)

Still allowed in novel forms

Modeling goals

- Build the *simplest* model that can help us understand the conditions required for change
- Model change as multi-generational acquisition: what generalization would each successive generation learn?

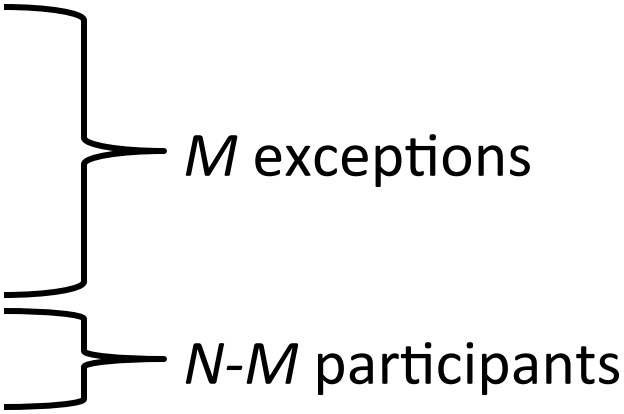
The productivity criterion

Productivity and generalization (Yang, 2005)

- What makes a worthwhile generalization? Real-time processing criterion
- A rule *R* can tolerate *M* exceptions if treating them as exceptions leads to an lower expected processing time than just memorizing everything

Exception lookup

IF form == x THEN x'
ELSE IF form == y THEN y'
ELSE IF form == z THEN z'
…
else DEFAULT



- Mathematically:
  - *N*- # of items that meet structural description of rule
  - *M*- # of items that meet structural description of rule but are exceptions
  - Criterion: $M < N / ln(N)$ assuming a Zipfian world (see Yang 2005 for proof)

Wide applications

- Productivity reigns (Yang *et al.*, 2012)
  - Paradigmatic gaps, no-default systems, overregularization

$M < N / ln(N)$?

The phenomenon

Stratal-cyclic models

- We see *overapplication* in cases like *sɪŋ.ɪŋ* and *lɔŋ.ɪʃ*
- On surface, */g/* not in coda → evidence of earlier application (Bermúdez-Otero, 2011):

  [sɪŋg][ɪŋg]
  [sɪŋ][ɪŋ]
  sɪŋ.ɪŋ

- Under stratal-cyclic models (e.g., Lexical Phonology, Stratal OT):
  - Morphological and phonological operations are interleaved
  - Phonological processes can apply at several levels/strata:
    - Stem, word, phrase

The change (Bermúdez-Otero, 2011)

	Stage 0	1	2	3
*elo**ng**ate*	ŋg	ŋg	ŋg	ŋg
*prolo**ng**-er*	ŋg	ŋg	ŋg	ŋ
*prolo**ng** it*	ŋg	ŋg	ŋ	ŋ
*prolo**ng*** ‖	ŋg	ŋ	ŋ	ŋ

Step 1 Step 2

Modeling results

Simulation

- Assembled all US English CHILDES data, transcribed using CMUDict
  - 2.8 million tokens in total
- Goal of simulation is to evaluate whether productivity can explain progression of change
- Hypotheses:
  - Step 1: level of ambiguity between **phrase/word** levels…
  - Step 2: level of ambiguity between **word/stem** levels…
  - …are high enough that reanalysis will occur

Step 1

- Innovation at the word level:

	phrase-level /*g*/-deletion (conservative)	word-level /*g*/-deletion (innovative)
$[_{PL}[_{WL}$ sin*g*-er]]	*g*	*g*
$[_{PL}[_{WL}$ sin*g*] $[_{WL}$ aloud]]	*g*	∅
$[_{PL}[_{WL}$ sin*g*]]	∅	∅

- If this change is to proceed, number of exceptions to a word-level deletion rule must not exceed tolerance
- Source of apparent exceptions is resyllabification preventing deletion

First problem: counting types and tokens

- Productivity is traditionally computed over *types*, unique words in the input
- However, in this case we see variation in *tokens*; each occurrence of *sing* can be different
- Some baseline strategies from dealing with this:
  - Conservative: a word type is an exception if it ever doesn't participate
  - Aggressive: a word type is a participant if it ever participates
  - Cautious: only count types that are completely consistent, e.g. always participate or never participate

Syllabification, I thought we were friends

- Unrestricted phrase level resyllabification prevents a productive generalization:

	Participants	Exceptions	Tolerance
Conservative	378	**821**	169
Aggressive	1002	**197**	169
Cautious	378	**197**	90

- But do we have evidence of restrictions?

A closer look at Elphinston's formal register

- "Upon solemn occasions […] if either feebly commence the word following" (formal register):
  - sin[*g*] aloud, prolon[*g*] it, stron[*g*] and mighty, sprin[*g*] eternal
  - Given as equivalent to word/stem level cases
- "But in different words it must indeed be a very strong, though not an impossible articulation, which expresses a final *g* before an initial *l* or *r*"
  - youn[*g*] Leander, lon[*g*] repose
- Analysis:
  - Always require "feeble" (unstressed) following syllable
  - Potential restriction to creating onset, not maximizing

Restricted phrase-level resyllabification

- With both stress and no-maximization restrictions:

	Participants	Exceptions	Tolerance
Conservative	378	671	150
Aggressive	1002	**147**	163
Cautious	378	147	83

- Summary: restrictions on phrase-level resyllabification were essential to change proceeding

Step 2

- Innovation at the stem level:

	word-level /*g*/-deletion (conservative)	stem-level /*g*/-deletion (innovative)
$[_{PL}\,[_{WL}\,[_{SL}$ sin*g*]-er]]	*g*	∅
$[_{PL}\,[_{WL}\,[_{SL}$ sin*g*]] $[_{WL}$ aloud]]	∅	∅
$[_{PL}\,[_{WL}\,[_{SL}$ sin*g*]]]	∅	∅

- If this change is to proceed, number of exceptions to a stem-level deletion rule must not exceed tolerance
- Source of apparent exceptions is suffixed stems

Step 2

- Easy transition, no matter how you count:

	Participants	Exceptions	Tolerance
Conservative	1074	**77**	163
Aggressive	1083	**68**	163
Cautious	1074	**68**	162

- *M* far below tolerance predicts that word level application without stem level will be unstable and rapidly change
  - Consistent with no account of a stable period

Predictions

- In languages with more aggressive phrase level resyllabification, processes will have difficulty moving from phrase to word level
- In languages with fewer bare stems surfacing, processes will not progress to the stem level at all
  - Dutch final coda devoicing (Booij,1997), Spanish nominals (Bermúdez-Otero, in press)
- Further test cases needed: phrase level and word level rules that stay where they are

Conclusions

- Gives first mechanistic account of how such a change can proceed
- Predicts that languages with different levels of domain ambiguity and different syllabification restrictions will allow different changes
- For this change to have happened, the learner must have relatively eager to reanalyze
- Future work needed to explore:
  - Learner's strategy regarding conflicting information for given word types: frequency?
  - Validity of predictions for other languages

Acknowledgments

- Many thanks to:
  - Gene Buckley
  - Ricardo Bermúdez-Otero
  - Charles Yang

Slides available at:
http://www.seas.upenn.edu/~lignos

Backup slides

(13)

	level	deletion?	*elongate*	*prolonging* ‖	*prolong it*	*prolong* ‖
a.	*Stage 0: Early Modern English*					
	SL	no	[iː.lɒŋ.geɪt]	[prə.lɒŋg] [ɪŋg]	[prə.lɒŋg] [ɪt]	[prə.lɒŋg]
	WL	no	[iː.lɒŋ.geɪt]	[prə.lɒŋ.gɪŋg]	[prə.lɒŋg] [ɪt]	[prə.lɒŋg]
	PL	no	[iː.lɒŋ.geɪt]	[prə.lɒŋ.gɪŋg]	[prə.lɒŋ.gɪt]	[prə.lɒŋg]
b.	*Stage 1: Elphinston's formal register*					
	SL	no	[iː.lɒŋ.geɪt]	[prə.lɒŋg] [ɪŋg]	[prə.lɒŋg] [ɪt]	[prə.lɒŋg]
	WL	no	[iː.lɒŋ.geɪt]	[prə.lɒŋ.gɪŋg]	[prə.lɒŋg] [ɪt]	[prə.lɒŋg]
	PL	yes	[iː.lɒŋ.geɪt]	[prə.lɒŋ.gɪŋ̸g]	[prə.lɒŋ.gɪt]	[prə.lɒŋ̸g]
c.	*Stage 2: Elphinston's casual register*					
	SL	no	[iː.lɒŋ.geɪt]	[prə.lɒŋg] [ɪŋg]	[prə.lɒŋg] [ɪt]	[prə.lɒŋg]
	WL	yes	[iː.lɒŋ.geɪt]	[prə.lɒŋ.gɪŋ̸g]	[prə.lɒŋ̸g] [ɪt]	[prə.lɒŋ̸g]
	PL	yes (vacuously)	[iː.lɒŋ.geɪt]	[prə.lɒŋ.gɪŋ]	[prə.lɒ.ŋɪt]	[prə.lɒŋ]
d.	*Stage 3: present-day RP*					
	SL	yes	[iː.lɒŋ.geɪt]	[prə.lɒŋ̸g] [ɪŋ̸g]	[prə.lɒŋ̸g] [ɪt]	[prə.lɒŋ̸g]
	WL	yes (vacuously)	[iː.lɒŋ.geɪt]	[prə.lɒ.ŋɪŋ]	[prə.lɒŋ] [ɪt]	[prə.lɒŋ]
	PL	yes (vacuously)	[iː.lɒŋ.geɪt]	[prə.lɒ.ŋɪŋ]	[prə.lɒ.ŋɪt]	[prə.lɒŋ]

Cost of storing exceptions (Yang, 2005)