Bruce Hayes
UCLA

Talk at Cornell University
March 29, 2018

Learning allomorphs as a basis for morphophonemic learning

I. BACKGROUND: LEARNING MODELS IN LINGUISTICS

1. The research program of generative grammar

- Chomsky (1965:24-26) and similar aspirational work, on human children:

Representative learning data → Learning device → Grammar → Elicitation and experimental testing

2. What is in the "learning device"?

- Learning mechanisms of some sort
- Linguistic theory; often construed as principles of Universal Grammar (UG)

3. The role of learning models

- Chomsky describes people, but we can try to *model* people by devising systems that perform the same way — linguistics as reverse engineering.[1]
- What would success (in the distant future) look like?
  - Machine-learned grammars will respond to elicitation identically to native speakers, including ambivalence.
  - They will perform identically in wug-tests and other experimental tasks.
  - Algorithms will occasionally learn *nonviridical* patterns (mismatch to ambient data), just as people do.
- I personally feel that a system as good as this is likely to be a reasonable analogue to the true system inside people's heads.

4. Phonology seems to be a good area for trying this sort of work[2]

- In Optimality Theory, we know how to rank constraints correctly, given a suitable set of inputs, winners, and losers (Tesar and Smolensky 1993, 2000)
- Free variation doesn't faze us, since several frameworks and ranking/weighting algorithms can accurately learn quantitative patterns presented to the learner (Boersma and Hayes 2000, Goldwater and Johnson 2003, Boersma and Pater 2016, etc.)

---

[1] Metaphor from Dupoux (2018).
[2] But syntax, too! I have noticed Clark and Lappin (2011) and Abend et al. (2017). Morphology: Ryan (2010).

- We can learn phonotactics from surface forms (Chomsky and Halle 1965: *brick, brick, bnick*); Hayes and Wilson (2008).

5. What has not yet happened (to my knowledge)

- A linguistic-based solver-system that can:
  - input **paradigm data** — like a classical phonology problem set, where phonological processes cause the morphemes to show up in different forms in different parts of the paradigm (cf. (9) below).
  - find the **underlying forms** of each morpheme present
  - **discover and order rules** (Chomsky and Halle 1968), or **rank a constraint set** in Optimality Theory (Prince and Smolensky 1993) so as to derive the surface forms[3]

6. Solving problem sets is an idealization/research way-station

- A large body of research tells us that the problem set answer falls far short of what real learners know about phonological patterning — there is just a lot more detail.
  - See Zuraw 2000, 2010, Albright 2002a, Albright and Hayes 2003, Ernestus and Baayen 2003, Hayes/Zuraw/Siptar/Londe 2009
- I have ideas on how to scale up the work here to handle detail, but will not go into this here.

II. A INCREMENTALIST APPROACH TO SOLVING PARADIGM PROBLEMS

7. What would be a sensible thing to try first?

- How do human analysts best solve these problems?
- I have a guess, and it appears in my textbook, *Introductory Phonology* (2008:Ch. 8): a "5-step program".

8. Toy illustration: Final consonants and clusters in Catalan

- Here is my pedagogical scheme as applied to five stems taken from Kenstowicz and Kisseberth's (1979:328) beautiful Catalan problem, (9) below.
- Quick answer:
  - Stems are /ultim/, /plen/, /klar/, /profund/, /fort/
  - Fem. sg. and fem. plur. endings are /-ə/ and /-es/
  - Phonology: /n/ and /r/ are deleted finally, after which /t,d/ delete / C ___], exposing new [n]'s and [r]'s to word-final position.

---

[3] Computer scientists, racing ahead of us, can already solve paradigm problems, most effectively with neural networks; see Cotterell et al. (2017). Finding systems that perform as well or better as these will be an important test of our theoretical ideas, perhaps leading us to revise them.

9. Step 1: divide the word into its morphemes

u l t i m |	‘last’	p r o f u n |	‘deep’
u l t i m | ə	‘last f.sg.’	p r o f u n d | ə	‘deep f.sg.’
u l t i m | e s	‘last f.pl.’	p r o f u n d | e s	‘deep f.pl.’
p l e |	‘full’	f o r |	‘strong’
p l e n | ə	‘full f.sg.’	f o r t | ə	‘strong f.sg.’
p l e n | e s	‘full f.pl.’	f o r t | e s	‘strong f.pl.’
k l a	‘plain’		
k l a r | ə	‘plain f.sg.’		
k l a r | e s	‘plain f.pl.’		

10. Step 2: Consulting the divisions made, list all allomorphs of each morpheme

‘last’:	[ultim]	f.sg.:	[-ə]
‘full’:	[plen] ~ [ple]	f.pl.:	[-es]
‘plain’:	[klar] ~ [kla]		
‘deep’:	[profund] ~ [profun]		
‘strong’:	[fort] ~ [for]		

11. Step 3: Consulting the allomorph list, find the segmental alternations

n ~ Ø, r ~ Ø, d ~ Ø, t ~ Ø

12. Step 4: Consider general hypotheses about underlying forms and reconstruct the derivations needed

- A good guess: *in Catalan nouns the feminine (prevocalic) forms always provide the right basis for the UR* (more on this below).
- So we can run with this idea and set up “proto-derivations”:

‘full’	*‘full-f.’*	*‘clear’*	*‘clear-f.’*	*‘deep’*	*‘deep-f.’*	
/plen/	/plen-ə/	/klar/	/klar-ə/	/profund/	/profund-ə/	hypothesized underlying forms
pleØ	—	klaØ	—	profunØ	—	*what the phonology must do*
[plen]	[plenə]	[kla]	[klarə]	[profun]	[profundə]	surface forms

- This set-up requires us to discover deletion at Step 5
- A full implementation of the approach would also try the other option, /ple/, /kla/, /profun/ with consonant insertion — and would fail at Step 5.

13. Step 5: Find a phonological grammar that will do what needs to be done

- For the above, in rule-based phnology, this would be:

N Deletion

$n \rightarrow \emptyset$ / ___ $]_{word}$

R Deletion

$r \rightarrow \emptyset$ / ___ $]_{word}$

Alveolar Stop Deletion

$\{t, d\} \rightarrow \emptyset$ / C ___ $]_{word}$ (must be ordered after N Deletion and R Deletion)

/plen/	/plen-ə/	/klar/	/klar-ə/	/profund/	/profund-ə/	hypothesized UR
pleØ	—	—	—	—	—	N Deletion
—	—	klaØ	—	—	—	R Deletion
—	—	—	—	profunØ	—	Alveolar Stop Deletion
[plen]	[plenə]	[kla]	[klarə]	[profun]	[profundə]	surface forms

- Success!
- Later we will do this problem again in Optimality Theory.

14. Can Steps 1-5, designed for humans, be made the basis of a learning algorithm?

- Step 1, "break up the words", turns out to be hardest, and is the focus here.
- Steps 2-3, "find the alternations", seems to be easier; see below.
- Step 4, "guess the UR's" is trivial for **Albrightianists**, hard for others; see below.
- Step 5, "find the grammar that can achieve these mappings", is easy if we are given OT constraints in advance, harder if we need to discover them.

- So let's outline the five steps and then solve some phonology problems …

15. What I am trying to avoid by using an incremental strategy

- **Explosion of hypothesis space**: the set of possible UR's coupled with grammars is incomparably vast for any decent-size problem.
- See Tesar (2014:§6.2), Jarosz (2015) for clear discussion on this point.
- My own system does have a big search-space bottleneck — Step 1 —but I think it's not *too* big.

16. The success of the strategy is a contingent matter

- Is it *really* possible to segment the words into morphemes without knowing the phonology?

17. A sobering example

- Thought experiment; you hear

[mapa] 'cat-sg.'	[maparu]	'cat-plur.'
[tapan] 'dog.-sg.'	[tapanu]	'dog-pl.'

- Is this /mapar/, with Final /r/ Deletion and suffix /-u/?

- ➢ Could be, since Catalan works like this.
- Is this /mapa/, /tapan/ with suffix allomorphs /-u/ post-C, /-ru/ post-V?
  - ➢ Could be, since Japanese works like this.
- We need to try out both cases, figuring out what principles could distinguish the two.

III. PUTTING IN THE HYPHENS — DEFINING THE TASK

18. The task

- Given a glossed paradigm, as above, discover an appropriate division into morphemes, without yet knowing the phonology.

19. Making it harder for realism's sake: discontinuous allomorphs

- Let include cases where the surface forms are discontinuous, due to
  - ➢ infixation, as in Tagalog [*bago*] ~ [*b*-um-*ago*].
  - ➢ metathesis of segments belonging to separate morphemes [pama], /naj-pama/ → [*na*-**p**-*j*-**ama**], as in Yagua (Powlison 1962)
- So we can't just put in hyphens; we must instead *coindex* every segment with the morphemic gloss of the morpheme it belongs to.

20. Sample of input data: part of my fictional "Suffix Fricatives" language

kuŋanpa	turtle-NOM	ruxiŋpa	dove-NOM	tuɸærpa	fox-NOM
kuŋanta	turtle-DAT	ruxiŋta	dove-DAT	tuɸærta	fox-DAT
kuŋanka	turtle-ACC	ruxiŋka	dove-ACC	tuɸærka	fox-ACC

piθoɸa	dog-NOM	ŋexeɸa	wolf-NOM
piθoθa	dog-DAT	ŋexeθa	wolf-DAT
piθoxa	dog-ACC	ŋexexa	wolf-ACC

- Pretty clear to any phonologist that we have:
  - ➢ stems: /kuŋan/, /ruxiŋ/, /tuɸær/, /piθo/, /ŋexe/
  - ➢ suffixes: /-pa/, /-ta/, /-ka/
  - ➢ /ptk/ spirantize intervocalically to [ɸθx].

21. Correct intended algorithm-output for Suffix Fricatives

k$_4$u$_4$ŋ$_4$a$_4$n$_4$p$_1$a$_1$	turtle$_4$ NOM$_1$	r$_5$u$_5$x$_5$i$_5$ŋ$_5$p$_1$a$_1$	dove$_5$ NOM$_1$	t$_6$u$_6$ɸ$_6$æ$_6$r$_6$p$_1$a$_1$	fox$_6$ NOM$_1$
k$_4$u$_4$ŋ$_4$a$_4$n$_4$t$_2$a$_2$	turtle$_4$ DAT$_2$	r$_5$u$_5$x$_5$i$_5$ŋ$_5$t$_2$a$_2$	dove$_5$ DAT$_2$	t$_6$u$_6$ɸ$_6$æ$_6$r$_6$t$_2$a$_2$	fox$_6$ DAT$_2$
k$_4$u$_4$ŋ$_4$a$_4$n$_4$k$_3$a$_3$	turtle$_4$ ACC$_3$	r$_5$u$_5$x$_5$i$_5$ŋ$_5$k$_3$a$_3$	dove$_5$ ACC$_3$	t$_6$u$_6$ɸ$_6$æ$_6$r$_6$k$_3$a$_3$	fox$_6$ ACC$_3$

p$_7$i$_7$θ$_7$o$_7$ɸ$_1$a$_1$	dog$_7$ NOM$_1$	ŋ$_8$e$_8$x$_8$e$_8$ɸ$_1$a$_1$	wolf$_8$ NOM$_1$
p$_7$i$_7$θ$_7$o$_7$θ$_2$a$_2$	dog$_7$ DAT$_2$	ŋ$_8$e$_8$x$_8$e$_8$θ$_2$a$_2$	wolf$_8$ DAT$_2$
p$_7$i$_7$θ$_7$o$_7$x$_3$a$_3$	dog$_7$ ACC$_3$	ŋ$_8$e$_8$x$_8$e$_8$x$_3$a$_3$	wolf$_8$ ACC$_3$

IV. THE PROPOSED APPROACH

22. Overall plan

- Let's use GEN + EVAL as in OT.
  - I.e., we lay out choices, then give formal criteria for picking a winner.
  - Such architectures are not just for linguistics, but are common in cognitive science as learning models.[4]

23. GEN and its size

- Our GEN = all possible coindexations of segments with the morphemes of their word.

  For: [sui] 'pig$_1$-VOC.$_2$'
  GEN is: { $s_1u_1i_1$, $s_1u_1i_2$, $s_1u_2i_1$, $s_1u_2i_2$, $s_2u_1i_1$, $s_2u_1i_2$, $s_2u_2i_1$, $s_2u_2i_2$ }

- (21) shows the correct coindexation for the Suffix Fricative language; I calculate there are 634 octillion others.
- In general, the GEN needed is really big, an issue below.

24. Choice of constraint-based model

- I use **Harmonic Grammar** (Smolensky 1986, Pater 2009, Potts et al. 2010, etc.), so constraints are weighted, not ranked.
- For the computations I use the **maxent** variant of Harmonic Grammar (Smolensky 1986, Della Pietra et al. 1997, Goldwater and Johnson 2003), for reasons to be made clear.

25. How we extract a prediction from the model

- The intended division of the words into morphemes will be the **highest-probability candidate**.
- In maxent, there is a tweaking procedure ("temperature"; Smolensky 1986:224-226) for weights that can make the winner-probability as close to 1 as we desire
- So for practical purposes coming in first by any margin is the equivalent of total victory.

26. Constraints

- The learning system will include a variety of constraints embodying what my experience suggests are properties shown by correct morpheme divisions.
- We will run through these constraints in what follows.

---

[4] See e.g. Samut (2010), entitled "Learning as search", in a computer science encyclopedia.

27. The more important basis for constraints: *similarity of allomorphs*

- Here is the right division for the Suffix Fricative language in (20):

A.

kuŋan-pa	turtle$_4$ NOM$_1$	ruxiŋ-pa	dove$_5$ NOM$_1$	tuɸær-pa	fox$_6$ NOM$_1$
kuŋan-ta	turtle$_4$ DAT$_2$	ruxiŋ-ta	dove$_5$ DAT$_2$	tuɸær-ta	fox$_6$ DAT$_2$
kuŋan-ka	turtle$_4$ ACC$_3$	ruxiŋ-ka	dove$_5$ ACC$_3$	tuɸær-ka	fox$_6$ ACC$_3$

B.

piθo-ɸa	dog$_7$ NOM$_1$	ŋexe-ɸa	wolf$_8$ NOM$_1$
piθo-θa	dog$_7$ DAT$_2$	ŋexe-θa	wolf$_8$ DAT$_2$
piθo-xa	dog$_7$ ACC$_3$	ŋexe-xa	wolf$_8$ ACC$_3$

- The right division yields this allomorph list:

*-pa ~ -ɸa NOM
*-ta ~ -θa DAT
*-ka ~ -xa ACC

[kuŋan] ‘turtle’, [ruxiŋ] ‘dove’, [tuɸær] ‘fox’, [piθo] ‘dog’, [ŋexe] ‘wolf’

- Here is a sample *wrong* division:

kuŋan-pa	turtle$_4$ NOM$_1$	ruxiŋ-pa	dove$_5$ NOM$_1$	tuɸær-pa	fox$_6$ NOM$_1$
kuŋan-ta	turtle$_4$ DAT$_2$	ruxiŋ-ta	dove$_5$ DAT$_2$	tuɸær-ta	fox$_6$ DAT$_2$
kuŋan-ka	turtle$_4$ ACC$_3$	ruxiŋ-ka	dove$_5$ ACC$_3$	tuɸær-ka	fox$_6$ ACC$_3$

*piθoɸ-a	dog$_7$ NOM$_1$	*ŋexeɸ-a	wolf$_8$ NOM$_1$
*piθoθ-a	dog$_7$ DAT$_2$	*ŋexeθ-a	wolf$_8$ DAT$_2$
*piθox-a	dog$_7$ ACC$_3$	*ŋexex-a	wolf$_8$ ACC$_3$

- The wrong division yields this allomorph list:

*-pa ~ -a NOM
*-ta ~ -a DAT
*-ka ~ -a ACC

kuŋan ‘turtle’ ruxiŋ ‘dove’ tuɸær ‘fox’

*piθoɸ ~ piθoθ ~ piθox ‘dog’
*ŋexeɸ ~ nexeθ ~ nexex ‘wolf’

- These allomorphs are *far less mutually similar* than the allomorphs of the correct answer.
- N.B. Pressure toward allomorph-similarity is known elsewhere in phonology (historical change, Kiparsky (1982); elicitation from children, Jo (2017); artificial grammar learning studies (Wilson 2006; White 2013, 2014); infixation locus (Zuraw 2007)).

28. SIMILARITY stated intuitively

- "*Penalize candidates to the extent that the allomorphs they imply fail to be mutually similar*."

29. Making this concrete

- Violations are based on the **summed dissimilarity of all allomorph pairs in the data** (all possible pairs of allomorphs of the same morpheme, summed across morphemes).
- English *visit* has three (in *visit*, *visitor*, *visitation*), so three comparisons would be made:

[vɪzət] ⟷ [vɪzəɾ]

[vɪsət^{h}]

- The calculation is based on **type frequency**, not token frequency (each allomorph counted just once, no matter how many words it appears in).
  - I've tried token frequency and it doesn't work as well.

30. Calculating (dis)similarity in phonology

- This is a long-studied problem and I'm following pretty standard methods.
- Let's break it down.

31. Breakdown

- The **similarity of two segments** is best modeled as a weighted sum of the features on which they differ (Wilson and Obdeyn 2009).
- The **weights of the features** can be obtained (White 2012) by setting them to fit a maxent model to the error rates in a segment-confusion experiment.
  - I used the confusion matrices from Cutler et al. (2004).
  - You also need a value for dissimilarity of segments to null ("MAX/DEP"), as in [klar] ~ [kla] , which I am obtaining by fitting to all of my learning simulations at once.
- The **similarity of two strings of segments** is the sum of the segmental differences under the best segment-by-segment alignment (Bailey and Hahn 2001, Albright and Hayes 2003).

32. Alignment of two strings

- Here is good alignment and a bad one for two of the *visit* allomorphs above:

```
a.  v  ɪ  z  ɪ  t          b. *v  ɪ  z  ɪ  t  ∅
    |  |  |  |  |              |  |  |  |  |  |
    v  ɪ  z  ɪ  ɾ              v  ɪ  z  ɪ  ∅  ɾ
```

- Given the segment-similarity values, there is a simple and clever recipe that finds the best string alignment (Sankoff and Kruskal 1999).

33. Summarizing the hierarchy of similarity-computation, bottom to top

Feature weights → Segment dissimilarity → String dissimilarity →
Aggregate dissimilarity of a candidate morpheme-division

34. First refinement to SIMILARITY: the stem/affix distinction

- Languages widely rank Faithfulness constraints for stems higher than for affixes (McCarthy and Prince (1995), Casali (1997), Walker (2011:§2.5)
- So we might expect it to be useful to penalize stem-variation more harshly than affix-variation.

  SIMILARITY$_{\text{STEM}}$

  "Penalize candidates to the extent that the stem allomorphs they imply fail to be mutually similar."

  SIMILARITY$_{\text{AFFIX}}$

  "Penalize candidates to the extent that the affix allomorphs imply fail to be mutually similar."

35. Second refinement to SIMILARITY: abstract away from size of data sample

- Stem similarity is normalized by dividing by the total number of stems.
- Affix similarity is normalized by dividing by the total number of affixes.

36. CONTIGUITY

- *Penalize a morpheme whose segments are not contiguous.*
- Invented (for phonology) by McCarthy and Prince (1995)
- In practice, this penalizes:
  - **Real cases** (infixation, metathesis across morpheme boundaries, Semitic morphology). So the weight of CONTIGUITY cannot be infinite.
  - **Stupid candidates** we want to rule out (quite common). Consider an example modeled on real-life Lomongo, with intervocalic /b/ Deletion. Correct parse:

'*duck*'	'*goose*'	
molon-e	baram-e	NOM.
molon-o	baram-o	ACC.
pa-molon	pa-aram	GEN.
ti-molon	ti-aram	DAT.

— Allomorphs of 'goose' are [baram] ~ [aram].

Wrong parse:

‘*duck*’	‘*goose*’	
molon-**e**	**b**-aram-**e**	NOM.
molon-**o**	**b**-aram-**o**	ACC.
pa-molon	pa-aram	GEN.
ti-molon	ti-aram	DAT.

— Wrong parse: ‘goose’ is [aram], [-e] and [-o] have circumfix allomorphs [b- -e] and [b- -o]; CONTIGUITY discourages them.

37. VARIEGATION

- This constraint is the least like ordinary phonology.[5]
- Consider the **Prefix-Temptation** language: /kimen/, /kurat/, /petep/, /loran/, with prefixes /ni-/, /bi-/, /ri-/ undergoing vowel harmony (vowel changes to [u] next to stem [u]).[6]
- Here is the bad analysis we want to avoid:

n-ikimen	‘sing 1p.’	n-ukurat	‘swim-1p.’
b-ikimen	‘sing 2p.’	b-ukurat	‘swim-2p.’
r-ikimen	‘sing 3p.’	r-ukurat	‘swim-3p.’
n-ipetep	‘sit 1p.’	n-uloran	‘think-1p.’
b-ipetep	‘sit 2p.’	b-uloran	‘think-2p.’
r-ipetep	‘sit 3p.’	r-uloran	‘think-3p.’

- This analysis is *perfect* with respect to SIMILARITY but is probably wrong.
- Such analyses can be discouraged by requiring the stem inventory to be Variegated.

38. VARIEGATION (intuitively)

- *“Disfavor candidate parses to the extent that the stem inventory is dominated by a single frequent initial or final segment.”*
- (37) is bad because every verb stem begins with either [i] or [u], an unlikely situation.
- I skip formalization for lack of time.

39. Summary: the full constraint set

- SIMILARITY$_{\text{STEM}}$
- SIMILARITY$_{\text{AFFIX}}$
- CONTIGUITY
- VARIEGATION

[5] As Paul Smolensky has pointed out to me, it is loosely related to the idea of the Rich Base in Optimality Theory (Prince and Smolensky 1993).

[6] Maltese Arabic, below, seems to be close to a real-life example where this danger arises.

40. Assigning weights to the constraints

- I gathered a big set of candidate parses for all my simulation-languages (see below), with correct and incorrect candidates.
  - Some incorrect candidates constructed by me, based on the point of the example.
  - Some from failed learning runs, or from the "beam" of a beam search (see below).
- I calculated a set of weights by which the correct morpheme division always receives the best harmony score.
  - Don't try to intuit the weights, since the violation scales are not comparable.

STEM SIMILARITY	12.2
AFFIX SIMILARITY	4.9
VARIEGATION	240.0
CONTIGUITY	10.8

- This relies on a huge virtue of Maxent: there is a provably reliable method for finding the weights that best fit the data. You can even perform the computation in Excel, as I did here.

41. Searching for the best candidate

- Now we have GEN (23) and EVAL (constraints and weights) — but how to find the winning candidate?
- Recall:
  - Candidates look like (21).
  - There are a huge number of them.
- Even worse, the search space is crammed with **local maxima** — the bane of learning-by-search.
- A few things have proven useful are given in (42).

42. Things that have helped

- Start by **finding nonalternating segments** and fixing their affiliations permanently.
- Then start the core search with random guesses.
- New candidates are found by **trying out edits on the old ones**.
- Use **beam search** — ten best candidates are kept in contention at once.
- Alternatingly search in various ways:
  - Search **small**: change affiliation of one segment at a time, or flip the)affiliations of two segments.
  - Search **big**: re-conceive the search space as the *list of allomorphs*. Change an allomorph, and implement the change throughout the data.

V. THE SIMULATIONS I TRIED AND THEIR RESULTS

43. Choice of data sets

- 10 are made-up languages, meant to pose some particular challenge to the system
  - the variegation language of (37).

   - an infixation language
   - Pseudo-Japanese, to test the Catalan/Japanese minimal pair described in (17).
- 10 are problem sets from Kenstowicz and Kisseberth (1979): Bizcayan, Chamorro, Catalan, Polish, Lamba, Maori, Maltese, Lomongo, Okpe, Modern Hebrew
   - These often include dramatic phonological phenomena of the type beloved to phonologists, e.g. in Okpe:

     [zũ] ‘fan’
     /ɛ-zũ-ɔ/ → [ezwõ] ‘fan-inf.’

     with opaque harmony of [ATR], [round] and [nasal].

44. Sample parsing result: Catalan

- Data: 27 adjectival stems, in masc. sg. (-Ø), masc. pl. ([-s]), f. sg. ([-ə]), f.pl. ([-es]).
- Sample issue:
   - We want [klarə] to be parsed [klar] + [ə], not *[kla] + [rə] (like Japanese).

45. Tableau

- Standard maxent calculations leading to probability values are given in bold.
- Candidates are parses of all 104 words in the problem, not just this stem.
- Just two candidates are presented; many others exist (they are bad, and lose)
- The bad candidate shown treats [r] as part of suffix, in Japanese fashion, in the three stems of the *klar-* pattern.

	SIMILARITY$_{STEM}$	SIMILARITY$_{AFFIX}$	VARIEGATION	CONTIGUITY			
weights:	12.2	4.9	240.0	10.8	*Harmony*	*eHarmony*	*p*
Correct parse	19.4	15.1	0.1481	0	347.8	$8.66 * 10^{-152}$	1
Parse treating stem-final [r] as if it were suffix	14.4	45.4	0.1481	0	435.3	$8.91 * 10^{-190}$	0

- Essential violations:
   - Correct analysis suffers from stem allomorphy: [kla] ~ [klar] etc., so worse on SIMILARITY$_{STEM}$
   - Bad analysis suffers from affix allomorphy: [-ə] ~ [-rə] and [-es] ~ [-res], so worse on SIMILARITY$_{AFFIX}$
- Intuitively, the stem violations are less salient because they occur only in a small subset of the total stem count (cf. (35)).

46. How well did the system do in general?

- All 10 made-up languages, plus 8 real languages: the outcome of the search is the correct answer.
- As we hoped, Catalan and Pseudo-Japanese (17) are each given their own correct analysis.
- For Okpe and Hebrew, search fails, landing on a bad parse. But this parse is far less harmonic than the linguist's parse, which I conjecture to be the true-but-unfindable optimum.

VI. THE REMAINING STEPS TO PHONOLOGY

47. Back to Step 2: list all allomorphs of each morpheme

- These are trivially read off the indexed representations used here, e.g. for (21) above:

$k_4u_4ŋ_4a_4n_4p_1a_1$	$turtle_4$ NOM_1	$r_5u_5x_5i_5ŋ_5p_1a_1$	$dove_5$ NOM_1	$t_6u_6\phi_6æ_6r_6p_1a_1$	fox_6 NOM_1
$k_4u_4ŋ_4a_4n_4t_2a_2$	$turtle_4$ DAT_2	$r_5u_5x_5i_5ŋ_5t_2a_2$	$dove_5$ DAT_2	$t_6u_6\phi_6æ_6r_6t_2a_2$	fox_6 DAT_2
$k_4u_4ŋ_4a_4n_4k_3a_3$	$turtle_4$ ACC_3	$r_5u_5x_5i_5ŋ_5k_3a_3$	$dove_5$ ACC_3	$t_6u_6\phi_6æ_6r_6k_3a_3$	fox_6 ACC_3

$p_7i_7\theta_7o_7\phi_1a_1$	dog_7 NOM_1	$ŋ_8e_8x_8e_8\phi_1a_1$	$wolf_8$ NOM_1
$p_7i_7\theta_7o_7\theta_2a_2$	dog_7 DAT_2	$ŋ_8e_8x_8e_8\theta_2a_2$	$wolf_8$ DAT_2
$p_7i_7\theta_7o_7x_3a_3$	dog_7 ACC_3	$ŋ_8e_8x_8e_8x_3a_3$	$wolf_8$ ACC_3

- For the morpheme glossed as NOM., index 1, we get the allomorph set { [-pa], [-ɸa] }

48. Step 3: Consulting the allomorph list, find the segmental alternations

- We use the standard procedure of string-alignment-by-similarity; already used in calculating SIMILARITY violations ((32)).
- Example: Polish [vut͡ɕ] 'lead-imp.sg.' ~ [vod͡z-e] 'lead-1 sg.'

a.

v	u	t͡ɕ
|	|	|
v	o	d͡z

not:

b.

v	u	t͡ɕ	∅
|	|	|	|
∅	v	o	d͡z

Therefore:

[u] ~ [o] is an attested alternation.
[t͡ɕ] ~ [d͡z] is an attested alternation.
[u] ~ [v] is not an attested alternation.

49. Step 4: Consider general hypotheses about underlying forms and reconstruct the derivations needed

- For the moment, let's all be Albrightianists.
- There is one privileged slot in the paradigm from which the UR of a stem is taken; the child makes this choice early and sticks with it for life.
- See Albright (2002a,b; 2005; 2008a,b; 2012) for supporting evidence.

- There are so few phonologically-distinct paradigm slots that it is quick and easy for an algorithmic learner to try them all.
- Catalan: we try the word final allomorph and the prevocalic (feminine) allomorph; as shown earlier only the latter will work since it encodes the essential underlying distinctions in stem-final consonants.

- If at some point I want to try more ambitious, trans-Albrightian, UR's, then we have a hidden-structure problem, and I hope to try out Jarosz's (forthcoming) promising new approach.

50. Step 5: Find a phonological grammar that will do what needs to be done

- Let's do it with OT.

51. We need a GEN

- Hooray, we know all the segmental alternations (Step 3), and we also have our candidate underlying forms.
- *No surface allomorph can exist that is not derived from a UR by attested alternations.*
- So, if we simply *apply every alternation in every possible location*, we will have a GEN that is in a sense "complete" and sufficient for learning.[7]

52. GEN example: German phonology

- Classical data: stems contrasting in final obstruent voicing, neutralized to voiceless in final position

  [rat] 'wheel-nom.' [rad-əs] 'wheel-gen.'
  [rat] 'advice-nom.' [rat-əs] 'advice-gen.'

- We segment morphemes, find allomorphs, and find one alternation, the [t] ~ [d] of 'wheel'.
- Activate GEN: using the extracted alternation [t] ~ [d], we replace the [t] of [rat] 'advice' with [d], obtaining a new allomorph *[rad]
  - ➢ This allomorph is completely impossible, yet it this turns out to be good, because when we use it as a candidate, it tells us how to rank phonological constraints (Pater et al. 2012).

/rat-əs/ 'advice-gen.'	IDENT(VOICE)	*INTERVOCALIC VOICELESS
☞ rat-əs		*
rad-əs	*	

  - ➢ This shows that German is a final devoicing language like Dutch, *not* an intervocalic-voicing language like Korean.

[7] This must be adjusted for epenthesis, which must be given a modest context to keep it from applying everywhere, or indeed an infinite number of times.

53. Constraints

- For now, I'm just being an unreconstructed classical OT person, typing in a rather thorough set of universal constraints (Markedness, Faithfulness).
- I am optimistic that these constraints can ultimately be learned from the data.

54. For each choice of UR's, do this:

- Concatenate the aligned allomorphs (either attested or GENerated) appropriately to form words, combining them in all possible ways to create a candidate set for each input.
- So far, I've found that this fits on a spreadsheet, not more than a few thousand rows…

55. The last step

- Add to your spreadsheet the constraints and their violations, and perform OT constraint ranking with any one of the many reliable ranking or weighting algorithms (see (4) above).
- I use the venerable Recursive Constraint Demotion algorithm (Tesar and Smolensky 1993 et seq.).
- If ranking converges, then the UR's you are testing are sufficient.

56. Sample from a machine-generated tableau for Catalan

- Choice of Albrightian UR for the stems: either of the Feminine forms (/ ___ -ə, -es).
- Candidates from GEN:
    - ➢ Catalan has [r] ~ [Ø] alternations like [klar-ə] ~ [kla], so GEN freely substitutes Ø for /r/.
    - ➢ Catalan has [k] ~ [ɣ] alternations elsewhere in the system, so GEN freely substitutes [ɣ] for /k/.

/klar/	ID(VOICE)	*CODA R	MAX
☞ kla			*
klar		*!	
ɣla	*!		*
ɣlar	*!	*	

57. Phonology problems solved so far in this way

- Bizcayan, Catalan, Chamorro, Lamba, Lomongo, Okpe, Polish, also Maori (not KK).
- Not Maltese, nor my concocted pseudo-Yagua (cf. KK 73-74), since both of these have metathesis and my GEN so far is can only concatenate morphemes.

VII. CONCLUSIONS

58. Incrementalism

- This is the key idea.

- Heuristics matching optimal behavior in phonologists keep the search space size under control, as we proceed through the Five Steps toward an answer.

59. An topic left unexplored here: non-phonological allomorphy

- Learning allomorphs is *necessary no matter what*, to learn irregular forms and non-phonological allomorphy — both quite abundant in languages.
  - I.e. even a full-scale effort to learn underlying forms and phonology together (dominant approach in the current literature) would still need to do something else to discover non-phonologically-derived allomorphs.
- My system can learn a system of lexical allomorphs found in Persian verbs.

60. Future work

- Remove the idealization given above in (6): scale up to include the highly detailed environments, used by human learners.
- With this done, go more empirical, with wug-testing:
  - Test the predictions of the learned grammars against intuitions of adult native speakers
  - Test the predictions of grammars learned on child-size dataset on the productions and intuitions of children.

——— *Finis* ———

References

Abend, Omri, Tom Kwiatkowski, Nathaniel J. Smith, Sharon Goldwater and Mark Steedman (2017) Bootstrapping Language Acquisition. *Cognition* 164: 116–143.

Albright, Adam. 2002a. Islands of reliability for regular morphology: Evidence from Italian. *Language* 78. 684-709.

Albright, Adam. 2002b. The identification of bases in morphological paradigms. Ph.D. thesis, University of California, Los Angeles. www.mit.edu/albright/papers/AlbrightDiss.html.

Albright, Adam. (2005). The morphological basis of paradigm leveling. In *Paradigms in phonological theory*, eds. Laura J. Downing, Tracy Alan Hall, and Renate Raffelsiefen, 17–43. London: Oxford University Press.

Albright, Adam. (2008a). Explaining universal tendencies and language particulars in analogical change. In *Language universals and language change*, ed. Jeff Good, 144–181. London: Oxford University Press.

Albright, Adam. (2008b). Inflectional paradigms have bases too: evidence from Yiddish. In *The bases of inflectional identity*, eds. Asaf Bachrach and Andrew Nevins, 271–312. London: Oxford University Press.

Albright, Adam. (2012). Base-driven leveling in Yiddish verb paradigms. *Natural Language and Linguistic Theory* 28: 475–537.

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

Apoussidou, Diana (2007) *The learnability of metrical phonology*. Utrecht: LOT.

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

Boersma, Paul (2001). Phonology-semantics interaction in OT, and its acquisition. *Papers in Experimental and Theoretical Linguistics* 6. Eds. Robert Kirchner, Wolf Wikeley & Joe Pater. Edmonton: University of Alberta. 24-35. [ROA 369, 1999].

Boersma, Paul, and Bruce Hayes. 2001. Empirical tests of the Gradual Learning Algorithm. *Linguistic Inquiry* 32:45–86.

Boersma, Paul and Joe Pater. 2016. Convergence properties of a gradual learning algorithm for Harmonic Grammar. In John McCarthy and Joe Pater, eds. *Harmonic Grammar and Harmonic Serialism*. London: Equinox Press.

Casali, Roderic F. (1997) Vowel elision in hiatus contexts: which vowel goes? *Lg* 73. 493–533.

Chomsky, Noam (1965) *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press.

Clark, Alexander and Shalom Lappin (2011) *Linguistic Nativism and the Poverty of the Stimulus*. Wiley-Blackwell.

Cotterell, Ryan, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden (2017). CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In CoNLL proceedings.

Dupoux, Emmanuel (2018) Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language learner. *Cognition* 173: 43-59

Ernestus, Miriam and Harald Baayen (2003). Predicting the unpredictable: Interpreting neutralized segments in Dutch. *Language* 79, 5-38

Goldwater, Sharon & Mark Johnson. (2003). Learning OT Constraint Rankings Using a Maximum Entropy Model. In Jennifer Spenader, Anders Eriksson & Östen Dahl (eds.), Proceedings of the Stockholm Workshop on Variation within Optimality Theory, 111–120. Stockholm: Stockholm University.

Hale, Kenneth L. (1973). Deep-surface canonical disparities in relation to analysis and change: an Australian example. In Sebeok, Thomas A. (ed.), *Current trends in linguistics 8: Linguistics in Oceania*, 401-458. The Hague/Paris: Mouton.

Hayes, Bruce and Colin Wilson. (2008). A maximum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry* 39: 379-440.

Hayes, Bruce, Kie Zuraw, Peter Siptár and Zsuzsa Londe. (2009). Natural and unnatural constraints in Hungarian vowel harmony. *Language* 85:822-863.

Jarosz, Gaja. (2006). Rich lexicons and restrictive grammars – maximum likelihood learning in Optimality Theory. Doctoral dissertation, Johns Hopkins University, Baltimore, Md.

Jarosz, Gaja (2015) Expectation driven learning of phonology. Ms. Department of Linguistics, University of Massachusetts, Amherst.

Jo, Jinyoung (2017) Learning bias of phonological alternation in children learning English. *Studies in Phonetics, Phonology and Morphology* 23(2):261-292.

Kenstowicz, Michael and Charles Kisseberth (1979) *Generative Phonology: Description and Theory*. San Diego: Academic Press.

Potts, Christopher, Joe Pater, Karen Jesney, Rajesh Bhatt and Michael Becker. 2010. Harmonic Grammar with Linear Programming: From linear systems to linguistic typology. *Phonology* 27: 77-117.

Pater, Joe. 2009. Weighted Constraints in Generative Linguistics. *Cognitive Science* 33: 999-1035.

Pater, Joe, Robert Staubs, Karen Jesney, and Brian Smith (2012) Learning probabilities over underlying representations. *Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology* (SIGMORPHON2012), pages 62–71, Montréal, Canada, June 7, 2012.

Powlison, Paul (1962) Palatalization portmanteaus in Yagua. *Word* 18:280-299.

Ryan, Kevin (2010) Variable affix order: grammar and learning. *Language* 86.4: 758-791.

Samut, Claude (2010) Learning as search. *Encyclopedia of Machine Learning*. Springer, 572-576.

Sankoff, David and Joseph Kruskal (1999) *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Stanford, CA: Center for the Study of Language and Information.

Smolensky, Paul (1986) Information processing in dynamical systems: Foundations of Harmony Theory. In James L. McClelland, David E. Rumelhart and the PDP Research Group. Parallel distributed processing: Explorations in the microstructure of cognition, Vol. 2: Psychological and biological models. Cambridge: MIT Press. 390-431.

Tesar, Bruce (2014) *Output-driven phonology*. Cambridge: Cambridge University Press.

Tesar, Bruce, and Paul Smolensky. (1993). The learnability of Optimality Theory: An algorithm and some basic complexity results. Ms. Department of Computer Science and Institute of Cognitive Science, University of Colorado at Boulder. Rutgers Optimality Archive ROA-2, http://ruccs.rutgers.edu/roa.html.

Tesar, Bruce, and Paul Smolensky (2000) *Learnability in Optimality Theory*. Cambridge, MA: MIT Press.

White, James (2013). *Bias in phonological learning: evidence from saltation*. PhD dissertation, UCLA.

White, James (2014). Evidence for a learning bias against saltatory phonological alternations. *Cognition* 130. 96–115.

Zuraw, Kie (2000). Patterned Exceptions in Phonology. Ph.D. dissertation, UCLA.

Zuraw, Kie (2010). A model of lexical variation and the grammar with application to Tagalog nasal substitution. *Natural Language and Linguistic Theory* 28(2): 417-472.