Learning allomorphs as a basis for morphophonemic learning

Bruce Hayes
Department of Linguistics
UCLA
Research question

- Suppose the right way to learn morphophonemics is to learn allomorphs first, before the phonology is known.
- How might this be done?
- And are there advantages to doing it this way?
The model abstractly characterized
Analytic task

- Devise a system that behaves like this:
- Given representative, glossed paradigms, it coindexes every segment with the morpheme that it belongs to.
Sample illustrative mini-language

- Person prefixes: /ni-/, /bi-/, /ri-/ alternating by Backness/Rounding Harmony
- Stems: /kimen/, /kurat/, /petep/, /loran/
- Morpheme labels are given an arbitrary index:
  
  nikimen sing$_4$-1p$_1$ nukurat swim$_5$-1p$_1$
  bikimen sing$_4$-2p$_2$ bukurat swim$_5$-2p$_2$
  rikimen sing$_4$-3p$_3$ rukurat swim$_5$-3p$_3$
  nipetep sit$_6$-1p$_1$ nuloran think$_7$-1p$_1$
  bipetep sit$_6$-2p$_2$ buloran think$_7$-2p$_2$
  ripetep sit$_6$-3p$_3$ ruloran think$_7$-3p$_3$
Correct intended result

- Model should assign indices to segments thus, indicating the morpheme they belong to.

\[
\begin{align*}
&n_{1i1}k_{4i4}m_{4e4}n_{4} \quad \text{sing}_{4-1p_{1}} \quad n_{1u_{1}k_{5}u_{5}r_{5a}a_{5}t_{5}} \quad \text{swim}_{5-1p_{1}} \\
&b_{2i2}k_{4i4}m_{4e4}n_{4} \quad \text{sing}_{4-2p_{2}} \quad b_{2u_{2}k_{5}u_{5}r_{5a}a_{5}t_{5}} \quad \text{swim}_{5-2p_{2}} \\
&r_{3i3}k_{4i4}m_{4e4}n_{4} \quad \text{sing}_{4-3p_{3}} \quad r_{3u_{3}k_{5}u_{5}r_{5a}a_{5}t_{5}} \quad \text{swim}_{5-3p_{3}} \\
&n_{1i1}p_{6e6e6e6} \quad \text{sit}_{6-1p_{1}} \quad n_{1u_{1}l_{7o}r_{7a}a_{7n}n_{7}} \quad \text{think}_{7-1p_{1}} \\
&b_{2i2}p_{6e6e6e6} \quad \text{sit}_{6-2p_{2}} \quad b_{2u_{2}l_{7o}r_{7a}a_{7n}n_{7}} \quad \text{think}_{7-2p_{2}} \\
&r_{3i3}p_{6e6e6e6} \quad \text{sit}_{6-3p_{3}} \quad r_{3u_{3}l_{7o}r_{7a}a_{7n}n_{7}} \quad \text{think}_{7-3p_{3}}
\end{align*}
\]

- N.B. segments might not be adjacent (metathesis, etc.)
A note on the task

- This is related to, but not the same as, as the task of unsupervised learning of words and morphemes (e.g., Goldwater et al. 2009).
  - I’m modeling late, not early, acquisition.
  - Meant to be 100% accurate, rather than heuristic.
  - Meant to feed into wug-testing (ability to synthesize new forms).
- Meant to be fed by unsupervised learning systems.
Coindexation as candidate selection

- GEN + EVAL, deployed with MaxEnt (Smolensky 1986, Goldwater and Johnson 2003)
- GEN = all possible coindexations of the segments
- EVAL uses various constraints of which the two main ones are as follows.
Faithfulness

- Adopt the well-motivated assumption (Kiparsky 1982, Steriade 2000, White 2016) that the final grammar should favor **minimization of allomorphy**.

- Let this be quantified thus: for each candidate, compile its “allomorph set” (like [ni-] ~ [nu-]) and assess violations of:

- **Faithfulness**: *penalize a candidate by the average dissimilarity of all allomorph pairs in the allomorph set.*
Calculating similarity

- I borrow the needed apparatus from:
- White 2016 (experiments $\rightarrow$ feature weights)
- Wilson and Obdeyn 2009 (feature similarity $\rightarrow$ segment similarity)
- Bailey and Hahn 2000 (similarity of best-aligned segments $\rightarrow$ string similarity)
Stem faithfulness

- Use separate, stronger version of Faithfulness for stems (McCarthy & Prince 1995, Beckman 1997)
VARIEGATION

Example of why this is needed; a segmentation that is *perfect* w.r.t. Faithfulness yet is wrong:

- 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.’
A constraint to enforce variegation in stems

- **VARIEGATION**
  - Penalize candidate assignments in which stems begin or end with only a few different segments
  - Measure employed: “worst case” count/total stems
Building a practical version of the model
20 Data Sets

- 10 are made-up languages, meant to pose some particular challenge to the system – such as the variegation language just given.
- 10 are problem sets from Kenstowicz and Kisseberth (1979): Bizcayan, Chamorro, Catalan, Polish, Lamba, Maori, Maltese, Lomongo, Okpe, Modern Hebrew.
Why problem sets?

- They include often-dramatic phonological alternations, intended to challenge the student.


- Such dramatic alternations are the stuff of phonological theory, and perhaps will help us offer a useful empirical challenge to our friends in computer science working on similar problems (Cotterell et al. 2017).
Weighting the constraints

- They are fitted (Excel Solver) to pick linguist-selected winning candidates, from a large spreadsheet of plausible rivals from the 20 sample languages.

- Rival candidates come from my own thinking, failed earlier versions of the model, and losing candidates from the search beam.
Searching for winners

- Start by fixing the affiliations of non-alternating segments.
- Then for the others, do a hill-climbing beam search.
- Search moves: (1) single-segment edits on the data set, alternating with (2) edits of the allomorph sets.
Criterion of adequacy

- Allocate high probability (> .99) to the linguist-selected correct answer.
How is the model doing?

- All 10 made-up languages, plus 8 real languages: the outcome of the search is the correct answer.

- Okpe and Hebrew: search fails. But its best candidate is far less harmonic than the linguist’s candidate, which I conjecture to be the true optimum.
The final step: full morphophonemic learning
Theme

- Full learning of the morphophonemic pattern becomes easier if you have the allomorph set in hand.
Begin by finding the alternations, with string-alignment

- String-aligning the discovered allomorph pairs, you can learn what the alternations are.

- English example:

  \[
  \begin{array}{c}
  \text{t} \\
  \text{r} \\
  \text{ŋ} \\
  \end{array} \quad [\text{t}] \\
  \quad [\text{r}] \\
  \quad \text{‘eating’}
  \end{array}
  \]

- Therefore: \([t] \sim [r]\) is an alternation.
Using the complete list of alternations

- You can use it to:
  - Rank the Faithfulness constraints (many can be immediately made undominated)
  - Construct a modest-sized yet comprehensive GEN function
- As a small-scale demo, let us use the allomorph sets to solve phonology problems.
Small-scale demo: full-UR and grammar learning, I

- First extract alternations from allomorph sets with alignment, as above.
- Then use this to make a defensible GEN: apply all alternations to all allomorphs.
- No winning candidate could ever be outside this set.
- No point in testing any other alternations.
Full-UR and grammar learning, II

- Code up a Universal Constraint Set for OT, following classical assumptions.
- (N.B. I and many people feel it would be better to learn the constraints instead.)
Full-UR and grammar learning, III

- Try out all possible combinations of attested allomorphs as UR’s, assuming the Single Surface Base Hypothesis (Albright 2002).
- The small GEN and small UR-set make it easy to check every choice.
Full-UR and grammar learning, end result

- Correct UR’s and ranking for Bizcayan, Chamorro, Catalan, Polish, Lamba, Maltese, Lomongo, and Okpe.

- This set is all the Kenstowicz/Kisseberth problems I tried that have purely-concatenative morphology (so, a simple GEN).
Planned next step: upgrade to wug-testing capacity

- Assertion: phonology problem set answers grossly underestimate the native speaker’s knowledge.
- There are hundreds of Islands of Reliability (Albright 2002) that must be learned as well; local environments where particular outcomes are favored (work of Ernestus/Baayan, Albright/Hayes, Zuraw, Becker, Gouskova, etc.)
- So the next step is to augment the model to discover the Islands (see Albright and Hayes 2003 for one effort).
- This will make possible empirical work; i.e. modeling the behavior of humans in wug-testing.
Conclusions

“Find the allomorphs first” seems to be a viable strategy for morphophonemic learning.

- It appears to be feasible.
- It likely has a smaller search space than alternative approaches (cf. Tesar 2014).
- It is needed for lexical allomorphy anyway (Paster 2006).
- It eases discovery of UR’s and constraint rankings.
Thank you

- Email advice/feedback welcome: bhayes@humnet.ucla.edu
References I

References II

References III