An Automated Learner for Phonology and Morphology

(and why it might be worth constructing)

Bruce P. Hayes
UCLA

1. Task

Given suitable paradigm data, prepare an algorithm (program) that will learn the morphological and phonological rules applicable within the paradigm.

2. Simple Example

- Language is “Pseudo-Finnish”, which may remind you of Finnish but is much simpler.
- See (5) for data.
- Part of paradigm learned: if given a present tense verb, provide the past tense.

3. Crucial Assumption About Pseudo-Finnish

- The sequences *[ei] and *[ti] are ill-formed (do not occur, are rejected by speakers in nonce forms, are hard for speakers to pronounce.)

4. Basic Rules and Generalizations

- The past is formed from the present by suffixation of /-i/ (see forms (a)-(n) on next page).
- If stem ends in /e/, as in plain-boxed forms (o)-(z): suffixation of /-i/ creates an /ei/ sequence, which is resolved by loss of /e/. Thus: /viljele+i/ \rightarrow [viljeli].

  **Hiatus Resolution:** e \rightarrow \emptyset / ____ i

- If stem ends in /t/, as in shaded (aa)-(dd): suffixation of /-i/ creates a /ti/ sequence, which is resolved by Spirantization of /t/. Thus /myyt+i/ \rightarrow [myysi].

  **Spirantization:** t \rightarrow s / ____ i

* Though I’m giving the talk, the work described is an equal collaboration with Adam Albright of UCLA.

i This assumption is not true of real Finnish.
• If stem ends in /te/, as in blackened (ee)-(jj), suffixation of /-i/ creates a /tei/ sequence, which is resolved by both Hiatus Resolution and Spirantization. Thus /tuunte+i/ → tuunti → [tuunsi].

5. The Pseudo-Finnish Data

<table>
<thead>
<tr>
<th>Gloss</th>
<th>Present</th>
<th>Past</th>
<th>Gloss</th>
<th>Present</th>
<th>Past</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. ‘sit’</td>
<td>istu</td>
<td>istui</td>
<td>s. ‘seek’</td>
<td>hake</td>
<td>haki</td>
</tr>
<tr>
<td>b. ‘talk’</td>
<td>puhu</td>
<td>puhui</td>
<td>t. ‘read’</td>
<td>luke</td>
<td>luki</td>
</tr>
<tr>
<td>c. ‘be visible’</td>
<td>näky</td>
<td>näkyi</td>
<td>u. ‘go’</td>
<td>mene</td>
<td>meni</td>
</tr>
<tr>
<td>d. ‘progress’</td>
<td>edisty</td>
<td>edistyi</td>
<td>v. ‘rise’</td>
<td>nouse</td>
<td>nousi</td>
</tr>
<tr>
<td>e. ‘want’</td>
<td>tahto</td>
<td>tahtoi</td>
<td>w. ‘tear’</td>
<td>repæise</td>
<td>repæisi</td>
</tr>
<tr>
<td>f. ‘loo’</td>
<td>katso</td>
<td>katsoi</td>
<td>x. ‘come’</td>
<td>tule</td>
<td>tuli</td>
</tr>
<tr>
<td>g. ‘be heard’</td>
<td>kuulu</td>
<td>kuului</td>
<td>y. ‘smile’</td>
<td>hymyile</td>
<td>hymyili</td>
</tr>
<tr>
<td>h. ‘ask’</td>
<td>kysy</td>
<td>kysyi</td>
<td>z. ‘listen’</td>
<td>kuuntele</td>
<td>kuunteli</td>
</tr>
<tr>
<td>i. ‘turn’</td>
<td>kæänty</td>
<td>kæänty</td>
<td>aa. ‘hum softly’</td>
<td>myyt</td>
<td>myysi</td>
</tr>
<tr>
<td>j. ‘stay’</td>
<td>vipy</td>
<td>vipyi</td>
<td>bb. ‘daub with honey’</td>
<td>luut</td>
<td>luusi</td>
</tr>
<tr>
<td>k. ‘grunt softly’</td>
<td>kakap</td>
<td>kakapi</td>
<td>cc. ‘sing in falsetto’</td>
<td>telt</td>
<td>telsi</td>
</tr>
<tr>
<td>l. ‘act silly’</td>
<td>katap</td>
<td>katapi</td>
<td>dd. ‘bathe ritually’</td>
<td>pinseet</td>
<td>pinseesi</td>
</tr>
<tr>
<td>m. ‘crawl’</td>
<td>turtos</td>
<td>turtosi</td>
<td>ee. ‘feel’</td>
<td>tunte</td>
<td>tunsi</td>
</tr>
<tr>
<td>n. ‘harbor wish’</td>
<td>kotek</td>
<td>koteki</td>
<td>ff. ‘cool by blowing’</td>
<td>huuhvete</td>
<td>huuhvesi</td>
</tr>
<tr>
<td>o. ‘cultivate’</td>
<td>viljele</td>
<td>viljeli</td>
<td>gg. ‘travel by llama’</td>
<td>amate</td>
<td>amasi</td>
</tr>
<tr>
<td>p. ‘look at’</td>
<td>katselo</td>
<td>katseli</td>
<td>hh. ‘circumambulate’</td>
<td>raute</td>
<td>rausi</td>
</tr>
<tr>
<td>q. ‘swallow’</td>
<td>nielaise</td>
<td>nielaisi</td>
<td>ii. ‘tickle with feather’</td>
<td>feete</td>
<td>feesi</td>
</tr>
<tr>
<td>r. ‘murmur’</td>
<td>solise</td>
<td>solisi</td>
<td>jj. ‘yearn inwardly’</td>
<td>vaante</td>
<td>vaansi</td>
</tr>
</tbody>
</table>

6. A Procedure for Getting Started on Morphology

• Construe the data pairs as **string mappings**:

\[
\begin{align*}
\text{[istu]}_{\text{present}} & \rightarrow \text{[istui]}_{\text{past}} \\
\text{[vipy]}_{\text{present}} & \rightarrow \text{[vipyi]}_{\text{past}} \\
\text{[katap]}_{\text{present}} & \rightarrow \text{[katapi]}_{\text{past}}
\end{align*}
\]

• Factor the mappings into **structural changes** and **conditioning environments**:

\[
\begin{align*}
\emptyset & \rightarrow i / [\text{istu}++]_{\text{present}} \quad (\text{“Insert i in final position after [istu]”}) \\
\emptyset & \rightarrow i / [\text{vipy}++]_{\text{present}} \\
\emptyset & \rightarrow i / [\text{katap}++]_{\text{present}}
\end{align*}
\]

• Collect similar cases, and generalize by replacing non-shared parallel strings with a variable:

\[
\emptyset \rightarrow i / [X++]_{\text{present}}
\]
• Expressing the result as a mapping between inflected forms, we obtain a **Morphological Mapping Constraint**:  

\[ [X]_{\text{present}} \rightarrow [Xi]_{\text{past}} \quad \text{“If the Present is } /X/, \text{ then the Past must be } /Xi/.” \]

7. **Where There is Phonology, Some Mappings Come Out as Separate**

a. /-e/ Stems (unshaded boxes in (3))

- \([\text{viljel}]_{\text{pres}} \rightarrow [\text{viljeli}]_{\text{past}}\) factorized: \(e \rightarrow i / [\text{viljel}____]_{\text{present}}\)
- \([\text{solis}]_{\text{pres}} \rightarrow [\text{solisi}]_{\text{past}}\) factorized: \(e \rightarrow i / [\text{solis}____]_{\text{present}}\)
- \([\text{hake}]_{\text{pres}} \rightarrow [\text{haki}]_{\text{past}}\) factorized: \(e \rightarrow i / [\text{hak}____]_{\text{present}}\)

Generalized to:

\[ [Xe]_{\text{present}} \rightarrow [Xi]_{\text{past}} \quad \text{“To form past of an /-e/ present, replace } [e] \text{ with } [i].” \]

This cannot be collapsed with the general constraint of (6), because it involves a different structural change.

b. /-t/ Stems (lightly shaded box in (3))

- \([\text{myyt}]_{\text{pres}} \rightarrow [\text{myysi}]_{\text{past}}\)
- \([\text{luut}]_{\text{pres}} \rightarrow [\text{luusi}]_{\text{past}}\)
- \([\text{telt}]_{\text{pres}} \rightarrow [\text{telsi}]_{\text{past}}\)

Will result in a mapping constraint:

\[ [Xt]_{\text{present}} \rightarrow [Xsi]_{\text{past}} \quad \text{“To form the past of a /-t/ present, replace } [t] \text{ with } [si].” \]

c. /-te/ Stems (black box in (3))

- \([\text{tunte}]_{\text{pres}} \rightarrow [\text{tunsi}]_{\text{past}}\)
- \([\text{huuvete}]_{\text{pres}} \rightarrow [\text{huuhvesi}]_{\text{past}}\)
- \([\text{amate}]_{\text{pres}} \rightarrow [\text{amasi}]_{\text{past}}\)

\[ [Xte]_{\text{present}} \rightarrow [Xsi]_{\text{past}} \quad \text{“To form the past from /-te/ present, replace } [te] \text{ with } [si].” \]

---

2 For earlier proposals on these lines, see Bochner (1993, 123), and work cited by Sproat (1992, 25-216).
PART II: WHERE DOES/CAN PHONOLOGY COME IN?

8. The Proposal

- Consider mapping constraints of high generality = their structural description matches many forms. Example above: most general mapping constraint is \([X]_{\text{present}} \rightarrow [Xi]_{\text{past}}\), with generality of 100%.
- Apply these mapping constraints everywhere.
- Where the answer is wrong, compare it with the right answer.
  
<table>
<thead>
<tr>
<th>viljele</th>
<th>*viljelei</th>
<th>should be</th>
<th>viljeli</th>
</tr>
</thead>
<tbody>
<tr>
<td>luut</td>
<td>*luuti</td>
<td>should be</td>
<td>luusi</td>
</tr>
<tr>
<td>tunte</td>
<td>*tuntei</td>
<td>should be</td>
<td>tunsi</td>
</tr>
</tbody>
</table>

- Examine a minimal window of segments surrounding the error point, and seek sequences that are illegal in the language, e.g.

  | viljel*[ei] |
  | luu*[ti]    |

Checking that the sequence really is illegal can be done on the spot, or else the determination of what strings are legal/illegal is completed prior to morphological analysis. From acquisition evidence, we suspect people do it the latter way.

9. Finding The Rules

- Develop mappings (phonological rules) that replace the phonologically-bad sequences with appropriate good ones.
- Observe that knowing the bad sequences is not by itself enough. You have to learn how the bad sequence gets fixed (structural change).
- So, explore, until you find the mappings that the language in question uses. Correct rules for Pseudo-Finnish:

  **Hiatus Resolution:**  \(/ei/ \rightarrow [i]\)
  
  **Spirantization:**  \(/ti/ \rightarrow [si]\)

- Apply these mappings freely and iteratively.

10. Promissory Note: Optimality Theory

- We have hopes of ultimately making the phonology Optimality-theoretic (see Prince and Smolensky 1993, and a vast literature since).
- Translation of our method: locate the Faithfulness constraint that must be demoted to allow /t+i/ to appear as [si] and /e+i/ to appear as [i].
11. Results of our Current Implementation on Pseudo-Finnish

- Original implementation programmed by BH, current greatly improved version programmed by Adam Albright.
- This program can learn Pseudo-Finnish. Specifically, it learns:
  - The default mapping \([X]_{\text{present}} \rightarrow [X]_{\text{past}}\)
  - A bunch of other, less general mappings, which are of no practical value but serve as way-stations on the way to the correct mapping.
  - The phonological rules Hiatus Resolution and Spirantization.

12. How We Test our Learner

We use the “Wug” test (Berko 1958):

- “If [kupy] were a present-tense verb, what would be its past?” “[kypyi]”
- “If [pupe] were a present-tense verb, what would be its past?” “[pupi]”
- “If [putut] were a present-tense verb, what would be its past?” “[putusi]”
- “If [mute] were a present-tense verb, what would be its past?” “[musi]”

13. Outline of the rest of the talk

- The theory of phonology assumed by the learner
- Applying the learner to German verbs: problems encountered, degree of success so far
- General discussion

THE THEORY OF PHONOLOGY IMPLICIT IN OUR LEARNER

14. Underlying Representations

- These play a minimal role.
- An “underlying representation” is simply the string projected from one inflectional form to another by means of mapping constraints alone.
- This “underlying representation” is modified by phonology to produce the correct surface form.
- Example: Mapping from surface \([tunte]_{\text{pres}}\), we obtain /tunte+i/, an “underlying representation,” which by phonology becomes surface \([tunsi]_{\text{past}}\).
- There is no conventional underlying representation serving as an abstract starting point for the whole paradigm.
15. What Do People Memorize in this System?

- At least enough forms of a paradigm to be able to project all the others by means of the mapping constraints/phonology.
- Where a paradigm includes phonological neutralization, then at least one paradigm member must be memorized that manifests the basic distinction unaltered.
- We are agnostic about just which forms are memorized, because there will almost always be multiple possibilities that work. Different speakers probably differ.
- At least some psycholinguists are happy with this view (e.g. Stemberger and MacWhinney 1988).

16. What About Exceptionful Phonology?

a) “Derived environment rules” (Kiparsky 1971). These should be easy to find: the target strings are those that are never found in the projected form unless also found in the form being projected from.

b) Lexical exceptions. See below.

c) Phonology triggered by specific affixes. We think the mapping constraints may already cover these. E.g. our mapping constraint (\[b\]) \[[x_t]_{\text{present}} \rightarrow [x_{s1}]_{\text{past}}\] represents a Spirantization triggered only by the Past tense ending /-i/. Real Finnish has this (Kiparsky 1973, 99).

MOVING ON TO HARDER CASES: THE GERMAN SIMULATION

17. Inputs

- 1643 German verbs, taken from pedagogical works.
- No prefixed verbs unless stem is bound.
- We focused on the following inflectional mapping:

  infinitive (e.g. [z\i\jo\=\o\n] ‘to sing’) → 1 plur. past ([z\i\jo\=\o\n] ‘we sang’)

18. The Regular Mapping

<table>
<thead>
<tr>
<th>Infinitive</th>
<th>1plur. past</th>
<th>Transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[\i\le\p\o=\o\n]_infin.</td>
<td>[\i\le\p\t\o=\o\n]_1plur. past</td>
<td>(\varnothing \rightarrow t / [\i\le\p__\o=\o\n]_\text{infin.})</td>
</tr>
<tr>
<td>[\i\de\k\o=\o\n]_infin.</td>
<td>[\i\de\k\t\o=\o\n]_1plur. past</td>
<td>(\varnothing \rightarrow t / [\i\de\k__\o=\o\n]_\text{infin.})</td>
</tr>
<tr>
<td>[\i\j\i\k\o=\o\n]_infin.</td>
<td>[\i\j\i\k\t\o=\o\n]_1plur. past</td>
<td>(\varnothing \rightarrow t / [\i\j\i\k__\o=\o\n]_\text{infin.})</td>
</tr>
<tr>
<td>[\i\t\a\i\I\o=\o\n]_infin.</td>
<td>[\i\t\a\i\l\t\o=\o\n]_1plur. past</td>
<td>(\varnothing \rightarrow t / [\i\t\a\i__\o=\o\n]_\text{infin.})</td>
</tr>
</tbody>
</table>
with many, many others, yield:

\[
[X\emptyset n]_{\text{infin.}} \rightarrow [X\emptyset n]_1 \text{plur. past}
\]

‘To form the 1st plur. past from the infinitive, insert /-t/ before the final /-\emptyset n/’

19. Some of the Phonology Engendered by the Regular Mapping

<table>
<thead>
<tr>
<th>Infinitive</th>
<th>1 plur. past, projected by regular mapping</th>
<th>correct 1 plur. past form</th>
<th>Phonology</th>
</tr>
</thead>
<tbody>
<tr>
<td>[glaub\emptyset n]</td>
<td>*[glaub\emptyset t\emptyset n]</td>
<td>[glaupt\emptyset n]</td>
<td>bt \rightarrow pt</td>
</tr>
<tr>
<td>[antvort\emptyset n]</td>
<td>*[antvort\emptyset t\emptyset n]</td>
<td>[antvort\emptyset t\emptyset n]</td>
<td>tt \rightarrow t\emptyset t</td>
</tr>
<tr>
<td>[ba\emptyset d\emptyset n]</td>
<td>*[ba\emptyset d\emptyset t\emptyset n]</td>
<td>[ba\emptyset d\emptyset t\emptyset n]</td>
<td>dt \rightarrow d\emptyset t</td>
</tr>
</tbody>
</table>

PROBLEMS THAT AROSE IN THE GERMAN SIMULATION: 1

20. A Triply-Ambiguous Structural Change

\textit{falten} ‘fold-infin.’ \rightarrow \textit{falteten} ‘fold-1 pl. pres.’:

- What is the structural change here?

\[
\begin{align*}
&\text{f\ a\ l\ t\ \emptyset\ n} \\
&\text{f\ a\ l\ t\ 0\ t\ \emptyset\ n} \quad \text{OR} \\
\end{align*}
\]

\[
\begin{align*}
&\text{f\ a\ l\ t\ \emptyset\ n} \\
&\text{f\ a\ l\ t\ 0\ t\ \emptyset\ n} \quad \text{OR} \quad \text{(this is more or less correct, modulo epenthesis)} \\
\end{align*}
\]

\[
\begin{align*}
&\text{f\ a\ l\ t\ \emptyset\ n} \\
&\text{f\ a\ l\ t\ 0\ t\ \emptyset\ n} \\
\end{align*}
\]

21. Keeping Temporary Wrong Hypotheses from Generating Nonsense

\[
\begin{align*}
&\text{f\ a\ l\ t\ \emptyset\ n} \quad \text{a wrong hypothesis from above (#1 in (20))} \\
&\text{f\ a\ l\ t\ 0\ t\ \emptyset\ n} \\
\end{align*}
\]

\[
\begin{align*}
&a\ \emptyset\ e\ l\ \emptyset\ n \quad \text{a reasonable hypothesis for} \\
&a\ \emptyset\ e\ l\ t\ \emptyset\ n \quad \text{angeln ‘to fish’~ angelt\emptyset n ‘we fished’} \\
\end{align*}
\]

\[
\begin{align*}
&X\ l\ Y \rightarrow \text{a disastrous generalization of the two:} \\
&X\ l\ t\ 0\ Y \quad \text{“Insert /t\emptyset l after /l, wherever it may be.”} \\
\end{align*}
\]
folgen

a truly embarrassing consequence of the disastrous generalization: Wug-test outcome for “folgen”

22. Current Idea for Avoiding This

- Forms with ambiguous structural changes (e.g. (20)) do not generate hypotheses.
- The mappings that generate such forms are ultimately learned—but only on the basis of clear input data; e.g. [tailön]infin. → [tailtön]1 plur. past.

GERMAN PROBLEM II: COMPLEX STRUCTURAL CHANGES


z i η t
gez uη oη

- What parsing procedure has the power to detect the two unchanged segments and locate three structural changes?
- We’re stuck on this for now, but suspect that the right route is to compare across paradigms as well as within them.

GERMAN PROBLEM III: COMPETING INCONSISTENT PATTERNS

24. Patterns in Irregular Verbs

- Many German verbs have irregular past tenses.
- The irregularity has a kind of near-regularity: irregular verbs fall into families.
- Usually, not all potential members of a family actually belong to the pattern:

<table>
<thead>
<tr>
<th>Irregular Family</th>
<th>Similar Regulars</th>
</tr>
</thead>
<tbody>
<tr>
<td>glaitön</td>
<td>gliton</td>
</tr>
<tr>
<td>raitön</td>
<td>ritön</td>
</tr>
<tr>
<td>jraitön</td>
<td>jritön</td>
</tr>
<tr>
<td>jtraitön</td>
<td>jtriton</td>
</tr>
<tr>
<td>arbeitön</td>
<td>arbeitatön</td>
</tr>
<tr>
<td>vaitön</td>
<td>vaitatön</td>
</tr>
</tbody>
</table>

- The more populous of these minor patterns are faintly productive (see below), so they are in competition with the regular pattern.

25. Goals

- Learn all the patterns, and establish a rough measure of their productivity.
- Make multiple, weighted guesses in Wug testing.
26. **Some Crucial Numbers to Keep Track of**

- The **scope** of a mapping constraint is the number of forms that *meet its structural description* (even if the mapping does not derive the correct mapped form).
- The **hit count** of a mapping is the number of cases it derives the correct form (abetted, sometimes, by phonology).
- The **reliability** of a mapping constraint = \((\text{hit count}) / (\text{scope})\).
- “Trustable” mappings have a reliability of 1 or close to it.
- Inferior, non-trustable mappings have a reliability near 0.

27. **Strategy For Wug Testing**

- Select the mapping constraint that has a reasonably good scope, and the highest reliability.
- Allow multiple guesses when reliability of rival processes is close.
- We leave open here what exactly is the scope/reliability tradeoff, but we think reliability is probably more important, for all but very small scopes.

28. **A Bit of Data for Checking**

- We submitted 9 made-up German verbs (infinitives) to two native speakers (HR, AM).
- Task: project the past tense form
- Instructions: treat them as authentic, native German verbs, not borrowed.
- Multiple guesses allowed
- Each guess rated on a scale of 1 (perfect) to 7 (terrible)

29. **The Learner’s Guesses Against Those of Native Speakers I**

First batch: there exists a mapping constraint with high reliability (at or near 1) that enforces the regular pattern. Crucial context string is shown below in boldface.

Result: both program and consultants adopt the regular pattern exclusively and confidently.

<table>
<thead>
<tr>
<th>Form</th>
<th>Guesses</th>
<th>Basis of Guesses</th>
<th>Rel.</th>
<th>Scope</th>
<th>HR</th>
<th>AM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bulieren</strong> ([büliːrən])</td>
<td>[büliːrən]</td>
<td>[Xiːrən] → [Xiːrən]</td>
<td>0.994</td>
<td>362</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>zefeln</strong> ([tsefəln])</td>
<td>[tsefəltən]</td>
<td>[Xəln] → [Xəltən]</td>
<td>1.0</td>
<td>177</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>tillern</strong> ([tilərn])</td>
<td>[tilərtən]</td>
<td>[Xərn] → [Xərtən]</td>
<td>1.0</td>
<td>148</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>zochen</strong> ([tsoχən])</td>
<td>[tsoχən]</td>
<td>[Xxən] → [Xxtən]</td>
<td>1.0</td>
<td>15</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
30. The Learner’s Guesses Against Those of Native Speakers II

Second batch: more than one mapping constraint competes to derive the past tense. We give all of the program’s guesses based on a reliability of greater than .3.

<table>
<thead>
<tr>
<th>Form</th>
<th>Guesses</th>
<th>Basis of Guesses</th>
<th>Rel.</th>
<th>Scope</th>
<th>HR</th>
<th>AM</th>
</tr>
</thead>
<tbody>
<tr>
<td>glingen (glïŋə̞ŋ)</td>
<td>[gliŋtə̞ŋ]</td>
<td>[Xən] → [Xtən]</td>
<td>0.876</td>
<td>1316</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>[gliŋə̞ŋ]</td>
<td>[Xliŋə̞ŋ] → [Xlaŋə̞ŋ]</td>
<td>1.0</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>krennen (krenə̞ŋ)</td>
<td>[krentə̞ŋ]</td>
<td>[Xən] → [Xtən]</td>
<td>0.876</td>
<td>1316</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>[krantə̞ŋ]</td>
<td>[Xənə̞ŋ] → [Xantə̞ŋ]</td>
<td>0.8</td>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>schneissen (ʃnaɪ̃sə̞ŋ)</td>
<td>[niastə̞ŋ]</td>
<td>[Xən] → [Xtən]</td>
<td>0.876</td>
<td>1316</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>[niisə̞ŋ]</td>
<td>[Xaisə̞ŋ] → [Xisə̞ŋ]</td>
<td>0.444</td>
<td>9</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>kreiben (krai̱bə̞ŋ)</td>
<td>[kraiptə̞ŋ]</td>
<td>[Xən] → [Xtən]</td>
<td>0.876</td>
<td>1316</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>[kriːbə̞ŋ]</td>
<td>(phonology: /bt/ → [pt])</td>
<td>1.0</td>
<td>4</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>bleiten (blai̱tə̞ŋ)</td>
<td>[blaitə̞ŋ]</td>
<td>[Xən] → [Xtən]</td>
<td>0.876</td>
<td>1316</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>[blitə̞ŋ]</td>
<td>(phonology: /tt/ → [tɾt])</td>
<td>0.667</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

31. Result Summary for Second Group

- All the program’s guesses are matched by at least one consultant-volunteered form.
- Every consultant-volunteered form is matched by program guess, generated by a mapping constraint whose reliability is greater than .3.

GENERAL DISCUSSION

32. Relation to Psycholinguistic Work

- This sort of thing is done by psycholinguists and connectionists all the time (mostly with English data); cf. Rumelhart and McClelland (1986), Daugherty and Seidenberg (1994), numerous other effort to model English past tenses.

- We think phonologists should participate too, since:
  → Phonologists know many interesting linguistic systems worth exploring.
  → There are quite a few theoretical devices developed by phonologists that may help learners quite a bit.
33. Relation to Phonological Work

To the extent that our current theoretical devices (help/hinder) learners, these devices receive an additional empirical test. Examples:

☞ Nonlocal rules. Learning umlaut, vowel harmony is hard, because the trigger of the process is “far away”.

Problem: for distant target, the number of possible strings to sift through becomes very large; too many hypotheses available for learning to be quick.

For umlaut, vowel harmony, the ability to scan only the vowels of a word would aid greatly, letting the learner zero in quickly on the crucial environment.

Existing phonological literature provides the right kind of mechanism; e.g. the “vowel projection” of Vergnaud and Halle (1979) or the “minimal scansion” of Archangeli and Pulleyblank (1994).

☞ We conjecture that ability to syllabify and classify syllables by weight would aid greatly in learning stress rules (see Dresher and Kaye 1990; and for an opposing view Daelemans et al. 1994).

☞ “Opportunistic” theories of underspecification (Steriade 1995): assign zero feature values to underlying forms post hoc, so that the elegance of the completed analysis is maximized. This seems like merely to expand the hypothesis space without leading us toward the right answer.

☞ We conjecture that approaches to phonology that use rule ordering, will get no support from the study of learning. Reason: any system that can handle the fearsome complexity of Latin declension classes would find the highly regular patterns of Yokuts or Tiberian Hebrew to be pretty trivial.

CONCLUSION: HANDCRAFTED GRAMMARS AND ALGORITHMICALLY-LEARNED GRAMMARS

34. Handcrafted Grammars

• In such grammars, the analyst’s full powers of imagination and insight are applied to shed light on the data pattern of a language, often with considerable elegance.

3 For recent defenses of serial derivations in phonology see Bromberger and Halle (1989) and McCarthy (1998). Tiberian Hebrew is defended as an argument for derivationalism by Idsardi (1997, 1998); derivationalism and Yokuts are discussed by Archangeli (1997).
35. Algorithmically-learned Grammars

- The analyst’s labors are directed toward algorithms that will arrive at workable analyses (which mimic native speaker behavior and intuitions) for any language.
- Once the algorithm is established, the grammar is a deterministic result of the algorithm and the input data.
- Thus the analyst models the speaker, with an algorithm; and the algorithm models the language.

36. Point I: What Should We Legitimately Strive to Model?

- People are like algorithms: they will learn whatever grammar their internal mechanisms and the external data dictate. We cannot intervene!
- Thus, hypotheses about grammars have a secondary, indirect character when compared to hypotheses about (human) learners. If we can make progress on the direct route, this seems preferable.

37. The Learnability Criterion in Generative Grammar

- Generative grammarians customarily, and rightly, marvel at the complexity of grammatical knowledge and how it can be arrived at by children so reliably and efficiently.
- Therefore, they set as an important criterion on theories that they help explain the learnability problem.

38. Point II: Handcrafted Grammars and Learnability

- Characteristically, they come with no procedure that tells you how the child could arrive at the grammar, given just the principles of the theory and the input data pattern.
- When the grammar is algorithmically learned, the learning theory that backs up the grammar is already in place.

39. Conclusion

- Handcrafted grammars will always be an important research activity—as a crucial heuristic.
- But because (a) they directly model what should be modeled; (b) they are fully responsive to the learnability question, algorithmically-crafted grammars are preferable in the long run.
References


Department of Linguistics
Box 154302
UCLA
Los Angeles, CA 90095-1543
bhayes@humnet.ucla.edu
www.humnet.ucla.edu/humnet/linguistics/people/hayes/hayes.htm