Linguistics 205 B. Hayes
Morphology Spring 2011

Problem Set #3: Selecting Bases with the Minimal Generalization Learner

Due Tues. 5/17 in class

1. Software support

- Bruce at 310 825-9507, bhayes@humnet.ucla.edu.
- Or bring your laptop to my office (office hours MW 11-12 and by appt.).

OBTAINING THE INPUT FILES

2. What it does

The software takes in a input file that looks like this:

Dative	*Nominative*	*Test forms:*	*Illicit sequences:*
zini]	ziu]	zini]	b]
malni]	malu]	malni]	g]
gorni]	goru]	gorni]	d]
nokni]	noku]	nokni]	
fikni]	fiku]	fikni]	
vagni]	vagu]	vagni]	
sagni]	sagu]	sagni]	
ragni]	ragu]	ragni]	
patni]	patu]	patni]	
bodni]	bodu]	bodni]	
sedni]	sedu]	sedni]	
lipni]	lipu]	lipni]	
tepni]	tepu]	tepni]	
ripni]	ripu]	ripni]	
zubni]	zubu]	zubni]	
vabni]	vabu]	vabni]	
sebni]	sebu]	sebni]	
tebni]	tewu]	tebni]	
lobni]	lowu]	lobni]	
kawni]	kawu]	kawni]	
ziwni]	ziwu]	ziwni]	

- It attempts to learn the mapping from the first form to the second, finding rules like "Peel off *-ni* and add *-u*." Or: "Peel off *-bni* and add *-wu*" (3rd and 4th to last forms).
- It collects statistics and evaluates all the rules that it learns.
- It (sometimes) uses the "illegal sequences" to learn simple phonological rules, which can be used to improve the performance of the morphological rules.
- Lastly, it takes a wug test, using its rules to make a variety of guesses for each test form.
  - In the present case, we simply wug-test all of the forms of the training set.

- The symbol "]" is a right-side word boundary. I didn't bother with the left side, where no phonology is found.

3. Downloading the problem files

- Download **ProblemSet3.zip** from the course web site, at http://www.linguistics.ucla.edu/people/hayes/205.
- Put it in a folder of your choosing. I suggest you make it (for now) a daughter of the top folder (c: on Windows computers). This keeps the commands short and avoids the problems that can arise in Java if there are spaces in your file specifications.
- Unpack the file. Most computers can handle .zip files already; and there is plenty of free software you can get by Googling "free unzip software" or the like.
- Once you've unpacked, you should see a bunch of nested folders.

4. Examine input files

- Take a look at the files inside these folders for a minute, just to get oriented. They all have the file suffix **.in**, for "input".
  - If you cannot see the suffix **.in** on your file names, fix this problem first. Visit http://www.linguistics.ucla.edu/people/hayes/120a/exe.htm and follow the directions there.
  - To open files with novel file suffixes such as **.in** in Windows, right click and select **Open With**. I suggest either Notepad or Wordpad, both of which come packaged with Windows; or a comparable plain text editor for Macs.
- As you can see, each file corresponds to a particular inflectional mapping, e.g. ACC → NOM. The files include the elements noted above:
  - the **learning data** for that particular direction
  - a **wug test**, in which the model is queried what it thinks the inflected form of the given base *ought* to be
  - a **phonotactics**, consisting a list of phonologically illegal sequences in this language. As you can see, it is a final-devoicing language.

RUNNING THE SOFTWARE

5. Getting Java

- You need to have the **Java runtime environment** on your computer.
- Search on your computer for **java.exe**. On my (Windows) machine it happens to be at C:\Program Files\Java\jre1.6.0_20\bin; yours is likely to be similar.
- If you don't already have the Java runtime environment, download it from:

  http://java.com/en/download/index.jsp

  and install it.

6. Running the Java program

- It should be part of the zipped package you downloaded. File name is MinimalGeneralizationLearner.jar.
- Click on it. If all goes well it will run.
- The Java program will cause a little interface to pop up. Click on "**Show all options**". Select "**Save rules**" and **de**select "**Use Features**".[1]
- Click the button **Open input file**, navigate the folders, find the **.in** file that you wish to analyze, and select it.
- Click **Learn morphology**.
- If all goes well, it will report its progress and write several files to the same folder in which the **.in** file was located. It should run very fast on a problem of this size.

*Run all six .**in** files* (each one is in its own folder).

THE PROBLEM

7. Data

We suppose a hypothetical language in which nouns are marked only for case. There are three cases, Nominative, Accusative, and Dative. You are given the paradigms for 21 stems:

Nominative	**Accusative**	**Dative**
zi	ziu	zini
mal	malu	malni
gor	goru	gorni
nok	noku	nokni
fik	fiku	fikni
vak	vagu	vagni
sak	sagu	sagni
rak	ragu	ragni
pat	patu	patni
bot	bodu	bodni
set	sedu	sedni
lip	lipu	lipni
tep	tepu	tepni
rip	ripu	ripni
zup	zubu	zubni
vap	vabu	vabni
sep	sebu	sebni
tep	tewu	tebni

[1] You can try features later on if you like. The feature file is included in the download package under the name **ProblemFeatures.fea**. You should copy the feature file and rename it to match the .in file that you're working with, e.g. ACCtoDAT.fea.

lop	lowu	lobni
kaw	kawu	kawni
ziw	ziwu	ziwni

As mentioned above, there is a rule of Final Obstruent Devoicing, which applies in a number of nominative forms.

Answer questions (a)-(p) below.

a. The accusative forms show an additional phonological rule that is *lexically sporadic* — in a traditional *SPE* analysis, you would diacritically mark the stems that undergo it. What is this rule?

b. In an Albrightian approach to inflectional mappings, if you had to pick a particular inflectional slot from which you were to derive the other two forms, which would you pick? Explain your answer.

The goal of the rest of the exercise is to see if a learning model can likewise come up with a reasonable answer to this question.

8. Inspecting your output files

- I assume that you have successfully run all six input files. (To locate the help center, see start of handout for help.)

d. Go to the **DATtoNOM** folder, and open **DATtoNOM.out**. Paste the contents of this file into your answer.

e. What is the phonology that was found? Is this an adequate characterization of the process?

- Note that the morphological changes are called "infixes" because I had to put in the boundary symbol "]" to get the software to work; hence the change is, strictly speaking, medial.

f. Now look at the morphological changes that were learned. The last three are not necessary. Why do you think they were learned? (If the answer is not clear, consult the Albright-Hayes readings.)

g. Now open and inspect the file **DATtoNOM.rules**, using a spreadsheet program such as Excel. Why does the rule ni $\rightarrow \varnothing$ / X ___] get 21 hits (despite forms like [vagni] ~ [vak])?

h. Even the perfect 21/21 rule, however, gets a **Confidence**, i.e. a final score, of only .955. Why is this so? (if you can't answer the question, do the Albright/Hayes readings).

i. Now open and inspect the file **DATtoNOM.sum**, using Excel or some other spreadsheet program. Sort it by Form ascending, Confidence descending. This will arrange the forms with the model's *best guess first*. How many times did the model get the right answer?

9. More output files: undoing Final Devoicing

Now go to the **NOMtoDAT** folder and open the file **NOMtoDAT.out**.

j. Why were no phonological rules found here?
k. What is the model's response to the phonological alternation that is giving it trouble?

Now open the file **NOMtoDAT.rules**.

l. Why is the performance of the simplest rule, $\varnothing \rightarrow$ ni / X ___], so bad?

m. Why does the best rule for the change k $\rightarrow$ gni get such a high score?

n. Give one rule which is a (modest) "island of reliability" for the regular change $\varnothing \rightarrow$ ni, and explain why it is an island of reliability.

10. Assessing the system as a whole

The goal now is to decide what would be the best Albrightian base on which to found the paradigm.

- For each of the six simulations, open the "xxx.sum" file (e.g. "NOMtoACC.sum") with Excel.
- Count the number of cases in which the model's best guess is the right answer. (For software assistance in this task, see this footnote. [2] Or do it very carefully by hand.)

---

[2] Insert two blank columns (**Alt i, c**) to the left of the column that has "by" in it.

Go to the input file and find the 21 forms of the learning data. Select them, return to the spreadsheet and paste them into two blank columns. Then delete the first of the two columns you pasted in (**Alt e, d**). You should now have a juxtaposition of the model's best guess with the correct answer.

Where necessary, edit the spreadsheet so that the rows are properly aligned. (copy a block, move it all down, paste).

You can now count the number of correct answers. This can be done by eye, if you like, or to do it perhaps a bit more safely:

Insert a blank column to the right of the "correct answer" column. Paste into cell H2 this formula:

=IF(ISBLANK(F2),"", IF(E2=F2, 1, 0))

Then copy it down the column.

Sum the cells you just entered (formula is "=sum([highlight the range])")
This is the score for this mapping.

p. Annotate the following diagram with the number of correct answers (0-20), and specify the base.