Interpreting sonority-projection experiments: the role of phonotactic modeling[1]

Bruce Hayes

Department of Linguistics, UCLA
bhayes@humnet.ucla.edu

ABSTRACT

A variety of experiments support the existence of "sonority projection": speakers represent severe sonority sequencing violations ([lba]) as less well-formed than modest ones ([bda]), even though neither is present in their native language. One interpretation of such findings is that the Sonority Sequencing Principle ([9]) is part of Universal Grammar. Modeling study, however, suggests that the results could be explained with more modest assumptions about UG. I use the Hayes-Wilson phonotactic learner ([6]) to show that sonority projection is possible in a system that merely attends to sonority differences and can generalize in the right way from the learning data.

Keywords: sonority sequencing, phonotactic learning, computational modeling.

1. INTRODUCTION

A standard observation of phonological typology is the principle of sonority sequencing ([9] et seq.): at each margin of the syllable, sonority normally declines going outward from the vowel. I will use the term "sonority projection" to describe cases in which speakers show knowledge of sonority sequencing that is not immediately deducible from the data they encounter in childhood; in Baker's terms ([2]), they "project beyond" the learning data to a richer pattern of knowledge.

Recent experiments suggest that sonority projection is real. Such experiments test native speaker reaction and behavior to clusters that are absent from the participants' native language, comparing severe violations of sonority sequencing such as [lba] against more modest ones like [dba]. Such experiments have gathered data involving (mis)perception ([1], [2], [3], [8]); accuracy of repetition ([1]), and well-formedness ratings ([5], [7]). They have generally found some sort of sonority projection effect. Some of the studies just cited have attributed sonority projection to the well-known hypothesis of Universal Grammar (UG)—the speakers are said to be able to project sonority sequencing in the absence of learning data because it is part of their inherent linguistic endowment.

Debates about UG are contentious, and the goal of this paper is not to weigh in on one side or the other. Instead, I adopt the working hypothesis that the experiments do implicate some sort of UG effect, then ask: *what is the UG that is needed*? While it is a logical possibility that the sonority sequencing principle is fully built into UG, I suggest that we can account for the experimental findings with more modest hypotheses about UG.

The tool I will use is computational modeling, specifically the phonotactic learning model of Hayes and Wilson ([6]). Deploying this model, I assess its ability to predict sonority projection effects using miniature UG's, none as rich as the full Sonority Sequencing Principle. As it turns out, the reason one can project sonority sequencing effects is that part of what is needed is already present in the learning data ([1]): the observed clusters that obey sonority sequencing serve as kernels for learning. Generalizing from them, one can obtain the full sonority pattern.

2. THE HAYES/WILSON PHONOTACTIC LEARNER

In my simulations, I fed phonotactically restricted "toy" languages to a version of the Hayes/Wilson learner (http://www.linguistics.ucla.edu/people/hayes/Phonotactics/). I also provided the model with various forms of UG. The learner then acquired constraint-based phonotactic grammars for these languages. I then tested the grammars on how they behaved for unheard clusters of varying degree of sonority violation to see if they had succeeded in projecting sonority. The Hayes/Wilson model is suited to this research for a variety of reasons. It can be provided with an *a* *priori* set of constraints that implement the UG under examination. The constraints penalize forms that violate them, ultimately lowering the probability assigned to them by the learned grammar. The model assigns weights to constraints following the criterion of maximum likelihood; i.e. the learned grammar assigns as much probability as possible to the observed data, insofar as this can be done with the given constraint set. Lastly, the learner employs a *Gaussian prior*, which acts as a force limiting the magnitude of individual weights, often forcing constraints to share their descriptive work with other constraints. This aspect of the model turns out be crucial for sonority projection.

3. SIMULATION I: THE BWA LANGUAGE

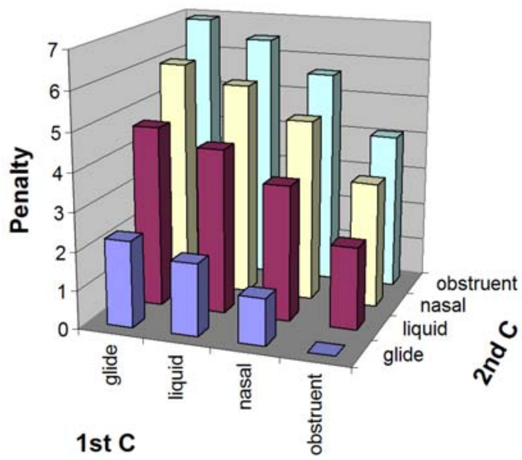
I begin with a pseudo-language I will call Bwa. Its phoneme inventory consists of the single vowel /a/ plus the consonants /ptkbgdfsvz, mn, lr, wj/. In the latter, the sequences separated by commas are the sonority classes I assumed: *obstruent*, *nasal*, *liquid*, *glide*. All words of Bwa are of the form [Ca] or [CCa], and the CC clusters are all and only those of the form *obstruent* + *glide* (e.g., [bwa]). I chose Bwa because its onset inventory resembles that of Korean and Mandarin,[2] two languages in which sonority projection has been demonstrated. Presumably, the simpler the onset inventory of a language, the harder it would be for speakers to project sonority without strong UG knowledge.[3]

Simulation: I based my feature system on that of Clements (1990), which expresses the sonority hierarchy in features: the features [sonorant], [approximant], and [consonantal] provides sonority cutoff points at the locations shown: *obstruents* | *nasals* | *liquids* | *glides*. For purposes of this simulation, vowels were not assigned sonority features. I set up a modest "UG": 32 constraints consisting of any sequence of two features that regulates sonority. I also included [syllabic], in order to express constraints that cover the full set of consonants. Here are examples: *[−consonantal][−sonorant] forbids *glide* + *obstruent* sequences, *[−sonorant][−consonantal] forbids *obstruent* + *glide* sequences, *[−syllabic] [−sonorant] forbids any obstruent-final cluster, and *[−sonorant] [−syllabic] forbids any obstruent-initial cluster. This UG only says, in effect, "care about sonority differences" — it is neutral between enforcing sonority sequencing and enforcing its exact opposite.

I fed the learning program these constraints along with a set of learning data from Bwa consisting of every legal word. The program assigned weights (often zero) to the 32 constraints, forming a grammar. I tested this grammar in a mode analogous to the experiments described above, querying it for the penalty score (dot product of weights and violations) it assigned to words such as [bwa], exemplifying all 16 possible sonority patterns that could arise in a two-consonant onset.

The result was sonority projection (Fig. 1). Syllables like [bwa], with the optimal sonority profile *obstruent* + *glide*, received a zero penalty score. Syllables like [wba], with the worst possible sonority profile, received the greatest penalty, and the other syllables received penalties that matched their degree of sonority violation.

Fig. 1: Sonority projection in Bwa



To understand how this happened, consider what clusters are penalized by particular sonority constraints. In Fig. 2, clusters penalized by *[+sonorant][−approximant] are shown above and to the left of the solid line; similarly for *[+approximant] [−syllabic] and the dotted line.

wb	lb	nb	bb
wn	ln	nn	bn
wl	ll	nl	bl
ww	lw	nw	bw

Fig. 2: The cluster types banned by two "sensible" sonority-regulating constraints.

In general, the 15 constraints enforcing "sensible" sonority order assign penalties for clusters in regions that go upward and leftward from any given point on the chart. The corresponding "non-sensible" constraints ban regions reaching downward and rightward from any given point.

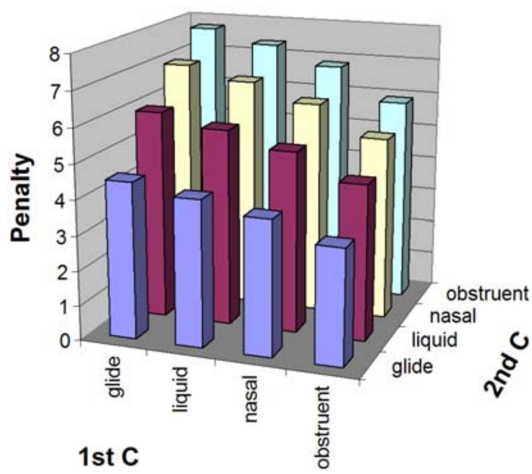
Consider next the maxent weighting. Every non-sensible constraint penalizes the one good cluster type of Bwa, *obstruent + glide.* As a result, the maxent weighting algorithm assigns these constraints a weight of zero—anything higher would produce a poorer fit to the data. What of the sensible constraints? The shortest observationally adequate grammar would consist of just two constraints: *[+continuant] [−syllabic], banning [nw] and all clusters above/leftward of it, and *[−syllabic] [+consonantal], banning *[bl] and all clusters above/leftward of it. But the Hayes/Wilson learner does *not* attempt to select the shortest grammar; in order to avoid overfitting it seeks an adequate grammar in which no one constraint has an especially high weight. Hence, it shares the descriptive burden by assigning positive weights to the 13 other constraints that regulate sonority in the sensible direction — which are also unviolated in the words of Bwa. The pattern that results is that the higher and further to the left a cluster is in Fig. 2, the more constraints penalize it. In the end, once the optimum weights and penalty scores have been computed, we obtain the sonority-projecting pattern of Fig. 1.

4. SIMULATION II: THE BA LANGUAGE

More ambitiously, we can inquire whether the learner could learn sonority projection even in a language that had no two-consonant onsets at all. The crucial point here is that vowels are even more sonorous than the most sonorous consonants. A syllable like [ba] has a very sharp rise in sonority—albeit not within a consonant cluster—and this can serve as a model for sonority projection just as the *obstruent + glide* clusters served as a model in Bwa. Indeed, the sonority fall between a consonant and a following vowel is sometimes regulated phonologically, as in Japanese, where only the most sonorous vowel [a] may occur after [w]. The idea, then, is to set up a learning simulation in which the sonority fall across CV serves a model from which sonority sequencing is projected in general.

For this reason, I modeled sonority learning in an even simpler language, Ba. Its vocabulary consists of all and only the CV words of Bwa. To model Ba, I altered the feature system in a simple way, giving the vowel phoneme /a/ the maximum sonority; in feature values [+sonorant, +approximant, −consonantal]. The UG was essentially the same as before. It emerged that sonority projection occurred in Ba as well; this is shown in Figure 2. In Ba, *all* consonant clusters are penalized, but the bad sonority violations are penalized more harshly than others.

Fig. 3: Sonority projection in Ba



5. POLYSYLLABLES AND CODAS: THE BABDA LANGUAGE

Languages that are less idealized than Bwa or Ba are harder to deal with for two reasons. First, when there are polysyllabic words, there will be an intervocalic singleton onset consonant, as in the [b] of [aba]. Here, the vowel plus consonant (i.e. [ab]) forms a falling sonority sequence, and this causes the simple learning procedure given above to fail. Beyond this, there is the fact that sonority sequencing is seen in both onsets and codas, and the preferred sonority direction is the opposite for each (rising vs. falling). Assuming, as seems likely, that future experiments show sonority projection for codas, this difference will have to be learned, and again it is beyond the capabilities of the model just given.

If the approach taken here is to extend to polysyllables and to sonority in codas, a richer UG will be needed—one that would specify the domains over which sonority sequencing is computed. I conjecture that this domain is the consonant cluster plus a "chunk" of the tautosyllabic vowel. This is perhaps plausible in the sense that the tautosyllabic vowel typically provides the crucial external acoustic cues to the identity of the consonant ([10]). Thus, in a hypothetical syllable like [plarp], the domains we want to consider are [p, l, *1st part of* a], [*2nd part of* a, r, p].

My final simulation pursued this idea. I constructed Babda, a language that has words of one to three syllables, where the syllables may be either Ca or CaC; with the segments being the same as in Bwa and Ba. With a small program I generated a random list of 5000 Babda words satisfying these criteria. In the file comprising these words, consonants were marked with an ad hoc diacritic Onset or Coda, according to their syllabic position. The feature system was the same as in Ba, except that the feature [±coda] was added in order to distinguish syllabic position. The constraint system was a doubled version of that for Ba, one half of which included the feature [−coda] in each feature matrix, the other half including [+coda]. The vowel [a] received a "split" representation, consisting of the unit [a_{onset}] followed by the unit [a_{coda}], each bearing the appropriate feature value for the feature [coda].

As should be clear, this procedure essentially cloned the pattern of Ba, but in two versions, one run "forwards" for onsets and the other in reverse for codas. Not surprisingly, when I tested the grammar on novel clusters—the same test as before, except with a mirror image test added for the codas—it projected sonority in the expected way.

6. CONCLUSION

The simulations here projected sonority using devices that are arguably more economical than stipulating the complete principle of sonority sequencing. The needed elements were: a feature system that characterizes the sonority continuum (including that of vowels) with cutoffs, a constraint set that uses these cutoffs to detect sonority sequencing, and a Gaussian prior that cautiously refrains from attributing all explanation to a single constraint. In the polysyllabic case, we must also somehow characterize the domains that are inspected for sonority differences (the same would be true for an a priori sonority hierarchy approach). What worked here was to split the syllable into two regions, each corresponding to an onset or coda together with the region of the adjacent vowel that provides the most effective cues.

7. REFERENCES

[1] Albright, Adam (in progress) Natural classes are not enough: Biased generalization in novel onset clusters. Ms., Department of Linguistics and Philosophy, MIT. http://web.mit.edu/albright/www/.

[2] Baker, C. L. (1979) Syntactic theory and the projection problem. *Linguistic Inquiry* 10:533-581.

[3] Berent, Iris, Donca Steriade, Tracy Lennertz, Vered Vaknin. (2007) What we know about what we have never heard: Evidence from perceptual illusions. *Cognition* 104: 591–630.

[4] Berent, Iris, Tracy Lennertz, Jongho Jun, Miguel A. Moreno, and Paul Smolensky (2008) Language universals in human brains. *PNAS* 105: 5321–5325.

[5] Daland, Robert, Bruce Hayes, James White, Marc Garellek, Andrea Davis, and Ingrid Normann (submitted) Explaining sonority projection effects. Ms., Department of Linguistics, UCLA.

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

[7] Pertz D. L. and T. G. Bever (1975) Sensitivity to phonological universals in children and adolescents. *Language* 51:149–162.

[8] Ren, Jie, Liqun Gao, and James L. Morgan (2010) Mandarin speakers' knowledge of the sonority sequencing principle. Paper given at the 20th Colloquium on Generative Grammar, University of Pompeu Fabra, Barcelona.

[9] Sievers, Eduard (1881). *Grundzüge der Phonetik*. Breitkopf und Härtel, Leipzig.

[10] Wright, Richard (2004). A review of perceptual cues and cue robustness. In Bruce Hayes, Robert Kirchner, and Donca Steriade (eds.) *Phonetically Based Phonology*, pp. 34-57. Cambridge: Cambridge University Press.

---

[1] Thanks to Paul Smolensky for suggesting that I try to model sonority projection in CV languages, and Robert Daland for astutely noticing that one of my models was already halfway there.

[2] The authors of the studies involved, [4] and [8], assume that Korean and Mandarin have no branching onsets at all; this is based on the assumption that in the observed *C + glide + vowel* sequences, the *glide + vowel* strings form syllable nuclei. However, this phonological analysis has not attracted consensus for either language ([5]). Thus it would be good for future research to test languages that unimpeachably lack branching onsets. I model a schematic language of this type in §4.

[3] For learning simulations that achieve sonority projection when trained on English data, see [5].