Testing phonological grammars with dictionary data

James Myers
National Chung Cheng University
Lngmyers@ccu.edu.tw

October 2, 2007
DRAFT [Comments welcome, and check back for updates]
[Based on talk presented at the International Workshop on Grammar & Evidence
http://www.ccunix.ccu.edu.tw/~lngproc/IWGE.htm]

1. Introduction

Most phonological research is actually corpus linguistics. This claim may sound odd, given that it is still relatively rare for theoretical phonologists to employ the sort of quantitative methods associated with corpus analysis (e.g. Frisch et al. 2004, Uffmann 2006). But think of the typical "problem set" in phonology class: the data aren't elicited native-speaker judgments of novel forms, as in syntax, but fixed sets of lexical items or phrases, usually taken from dictionaries. These data sets are corpora, since they are preexisting rather than generated through experimental manipulation. Phonologists have much to gain from experiments, whether they involve native-speaker judgments (e.g. Frisch and Zawaydeh 2001), phonetic measurements (e.g. Port and Leary 2005), or something else (e.g. Ohala 1986), but most phonological argumentation is still built on the reasonable assumption that corpus patterns reveal something about the mental grammar that went into producing them.

Despite the predominance of dictionary data in phonological research, there is still some confusion over how corpus-based arguments should work in the study of grammar. In this chapter I hope to clarify the logic underlying phonological argumentation from dictionaries, and then show how the logic can be formalized as a set of simple computerized procedures. Unlike several recent attempts to automate phonological grammar analysis (e.g. Tesar and Smolensky 1998, Boersma and Hayes 2001, Hayes and Wilson to appear, Pater et al. 2007a), my proposal is not a learning algorithm. Instead, it is meant as a tool for testing the statistical reliability of hypothesized grammars proposed as part of a traditional phonological analysis. Thus my proposal is intended to build on and clarify traditional methods, rather than representing a new theory of phonological competence or performance.

I lay the groundwork in section 2 by describing how dictionary data should work in phonological argumentation. I then describe, in section 3, how quantitative analysis can be naturally incorporated into traditional phonological methodology. In particular, I show how

corpus data can be used to test grammatical hypotheses expressed in the framework of Optimality Theory (Prince and Smolensky 1993/2004), against the statistical null hypotheses that the proposed constraints or proposed ranking aren't necessary. Explicit instructions are given for implementing the statistical methods in the free statistical software R. Section 4 provides conclusions and considers future prospects of this way of looking at phonological methodology.

2. Corpus analysis in phonology

Because science is often naively thought of as inherently experimental, and since pronouncements against corpus analysis have been a staple of the generative literature from the beginning, it is necessary to explain why corpora are indeed valid data sources in the study of grammar. In this section I first distinguish three distinct roles played by corpus analysis, only one of which is relevant here. I then show that despite well-known arguments in the generative literature, corpus frequency does indeed correlate with grammaticality. Finally I show that corpus analysis, despite its reputation as empiricist, actually requires a healthy dose of rationalism as well.

2.1 Corpora as performance evidence for competence

The goal of this chapter is to understand how phonologists can test grammatical hypotheses against dictionary data. Though this is the most common way corpus data are used in phonology, it must be contrasted with two other goals that many researchers seem to think are more prototypical of corpus analysis. Namely, some researchers think of corpus analysis as concerned solely with the study of performance, not competence, while others focus on the use of corpora, not as data for testing hypotheses, but as input to grammar learners. In this section I clarify the distinctions among these three goals of corpus analysis, and explain how the goal of using corpora as evidence for grammar may be justified.

One goal guiding corpus analysis is the description of performance for itself. This approach is founded on the correct observation that corpus data represents performance rather than competence. Chomsky (1957:15) pointed out that "the set of grammatical sentences cannot be identified with any particular corpus of utterances," and the same holds for a dictionary of words. Whether or not a word ends up in a dictionary depends on many factors other than grammar, including historical accidents (borrowings and the invention and loss of concepts) and processing constraints (words can't be too long nor too numerous). But these truisms do not mean that corpora can only be used to study performance itself. A corpus is the record of successful applications of grammar by native speakers, so it would be very surprising if corpora showed no

trace of grammar at all, mixed in with those traces of historical accidents and processing constraints.

Another goal that may be adopted in corpus research is to use corpora as input to automatic grammar-learning algorithms. At least since Chomsky (1965), generativists have claimed that this task is impossible without the learner having considerable built-in knowledge, so contemporary learning models are meant not as general-purpose discovery procedures, but empirical proposals about how actual children are innately built to acquire language (e.g. Tesar and Smolensky 1998, Boersma and Hayes 2001, Hayes and Wilson forthcoming, Pater et al. 2007a).

This chapter, however, has a third goal, namely to use corpora as evidence for grammar. Given this goal, shifting the focus from the corpus to the learner misses the point, since a corpus is a record of adult language usage first, and only secondarily input to a child learner. Clearly, even adult phonologists, who have presumably lost access to any innate language-learning algorithms, are capable of testing grammatical hypotheses on the basis of dictionary data. After all, this methodology has been central in the arguments for virtually all constructs in theoretical phonology, from phonemes to Optimality-Theoretic constraints. Even Chomsky and Halle (1968), both fluent English speakers (one native), rely less on their own intuitions than on Kenyon and Knott (1944), a pronunciation dictionary.

Why should phonologists favor corpus analysis over the experimental data (elicited native-speaker judgments of acceptability) common in syntax? I've just mentioned one reason, namely the burden of tradition. Phonological theory is founded on well over a century of corpus-based analysis, and it would be highly impractical to rederive all of the basic concepts using experimental data.

Less trivially, even if the patterns in dictionaries actually reflect diachronic processes rather than synchronic grammar (Ohala 1986), these processes are also systematic and psychological, and arguably are composed out of the same stuff that composes synchronic grammar (see Blevins 2004 for arguments against this, and Hayes and Steriade 2004 and Kiparsky 2006 for defenses of the traditional view).

Corpus data also make cross-linguistic comparisons much easier than would a reliance solely on judgment data. For this reason, phonologists have a much longer history of addressing typological issues than do syntacticians. Belying their titles, *The sound pattern of English* (Chomsky and Halle 1968) ranges far beyond English, whereas *Aspects of the theory of syntax* (Chomsky 1965) is almost exclusively restricted to aspects of *English* syntax. It's not surprising, then, that Optimality Theory, the currently dominant theory in phonology, is so typologically oriented.

Finally, given the dual status of a corpus as output of language use and input to language

learning, shifting to judgment-based analyses would not avoid the necessity of corpus analysis. Speakers cannot judge words or nonce forms without checking them for familiarity or comparing them with other words (e.g. Bailey and Hahn 2001). In other words, speakers make phonological judgments at least partly by conducting a sort of naive corpus analysis. Researchers who don't anticipate them with formal corpus analyses of their own might misinterpret the effects of lexical frequency or superficial analogy for truly productive grammar.

The increasing use of experimental data in the testing of phonological hypotheses is certainly welcome; there's no such thing as too much data. Nevertheless, given the justified centrality of corpus data in phonological theory, it remains important to understand how they may best be analyzed.

2.2 Grammaticality and corpus frequency

Despite the solid reasons for testing phonological hypotheses with corpora, implicitly recognized throughout the history of phonology, corpus data cannot be treated as intrinsically "direct" any more than any other sort of performance data. In particular, whether or not a given form is attested in a corpus, and how frequent it is if it is attested, is logically independent of its grammaticality. Nevertheless, the traditional anti-corpus rhetoric in the generative literature overstates the case. In this section I explain why, and show how attestation and frequency actually provide very useful information about grammaticality.

The nonidentity of grammaticality with corpus attestation and frequency follows immediately from the recognition of the latter as mere performance data, affected by numerous factors other than grammar. In lexical phonology the most important of these factors is rote memory. Lexical exceptions survive in a corpus, not by virtue of grammar, but in spite of it. Hence they are, strictly speaking, ungrammatical. Grammatical theory has no need to explain them, since they are already sufficiently explained by the mind's ability to memorize. Therefore, efforts such as those of Inkelas et al. (1997) and Pater (to appear) to develop a grammatical theory of exceptions may be missing the point, by obliging grammar to account for every aspect of performance.

Yet given their origins in language use by people who know their grammar, dictionaries must also record grammatical information as well. In particular, we expect there to be a correlation between the grammaticality of a form type (i.e. class of words defined by phonological structure) and its type frequency in the corpus. This assumption is no different in kind from the standard assumption of a correlation between grammaticality and native speaker judgments of acceptability.

The fact that this rather banal observation has sometimes been resisted by generative

linguists can be explained, I think, by confusion over the three different goals of corpus analysis described in the previous section. For example, when Chomsky (1957:16) says that grammaticality cannot be equated "with the notion 'higher order of statistical approximation'," or when Chomsky (2002:102) says that "[i]f you took a videotape of things happening out the window, it would be of no interest to physical scientists," the argument assumes that corpus linguistics necessarily involves either a non-analytic description of the corpus itself or, at best, automatic grammar learning of a highly naive sort. Yet corpus analysts who use observations to test prespecified hypotheses, rather than merely describing surface forms or building automatic grammar learners, entirely side-step these sorts of criticisms.

The logic runs like this. The only way a grammar can reveal its existence in a corpus is by frequency differences among competing form types, since a grammar that results in equal numbers of "grammatical" and "ungrammatical" forms cannot be detected. To take a schematic example, suppose a phonologist hypothesizes that structure [+X] is grammatical, and that [-X] is ungrammatical. A critic points out that examples of [-X] words actually exist in the language. A reasonable defense of the original hypothesis is still possible, however, if the number of [-X] words is "sufficiently low" to dismiss them as lexical exceptions.

In one of the rare discussions in the generative literature of the logic of corpus analysis, Duanmu (2004) describes the correlation between grammaticality and frequency in a more complex way. He acknowledges that the higher the frequency of a structure in a corpus, the more likely it is to be grammatical. Yet he suggests that for ungrammaticality, the inference runs in the opposite direction, from grammar to corpus; namely, the less grammatically well-formed a structure, the less likely it is to appear in the corpus. This apparently follows from the assumption that there is only one way for a form type to be common in a corpus, namely by being grammatical, but there are two ways for a form type to be rare, either by being ungrammatical (systematic gaps or exceptions) or by being grammatical but disfavored for some other reason (accidental gaps like historical accidents).

As a practical matter, however, there is no need to assume this asymmetry. First, there are also extra-grammatical ways for a form type to become common, such as superficial analogy and borrowing of lexical classes. Secondly, commonness and rarity are defined relative to each other, so the crucial factor is the difference in number between the two types, not the size of each category separately.

These considerations make it clear why quantitative analysis might be useful to the analysis of phonological corpora, as explained more fully in section 3. However, there is still one more basic issue to address before getting there.

2.3 Universals in phonological corpus analysis

Very likely the main reason for the confusion over the different goals of corpus analysis is the fact that a corpus is intended to contain observations collected with a minimum of a priori assumptions. This fools many linguists, on both sides of the rationalist/empiricist divide, into thinking that corpus analysis itself must be conducted with a minimum of a priori assumptions. However, this certainly is not how corpus analysis is usually employed in generative phonology, which adopts a decidedly rationalist, universalist approach. In fact, it can be argued, as I do in this section, that grammar-oriented corpus analysis cannot be done without making prior assumptions about what one is looking for.

To set the scene for the quantitative models in section 3, the best way to explain why is with a statistical example. The chance probability of choosing any particular card from a deck is 1/52. This means that no matter which card you choose, the result is so unlikely to have occurred by chance that it meets the usual criterion for statistical significance ($p = 1/52 = 0.019 < 0.05$). Yet clearly such a "finding" is meaningless without establishing ahead of time which card is expected. In a similar way, any set of *n* items in a corpus is just as unlikely to be selected for discussion as any other set of *n* items, and without prespecifying why one set represents a pattern and not the others (i.e. why the theorist considers them "grammatical"), there's no point trying to make an argument on *p* values alone.

In other words, to make the case that a corpus pattern is statistically significant, one first has to define what will count as a "success." There are essentially two ways to do this. One is more rationalist: even before looking at the corpus, declare what pattern is predicted. The other is more empiricist: look at the corpus, see what patterns seem to be present, and then test whether they are truly statistically significant (highly unlikely to have arisen by chance). In practice, both methods are often used in conjunction. For example, the more empiricist method is often used when studying a new phonological pattern in a previously unfamiliar language, but if the pattern can be expressed in terms of a universal principle, the more rationalist method is used to test the principle in other languages as well.

It might be objected that there actually is a single "correct" analysis of a phonological corpus, and that is the grammar acquired by the child exposed to it, using her innate biases. These innate biases may, in principle, cause the learner to ignore a statistically significant pattern in a corpus, and instead focus on something else. For example, some corpus could have equal numbers of [+X] and [-X] words, but the child may still innately know that [+X] must be grammatical and [-X] ungrammatical. Such considerations aren't implausible, but they certainly don't undermine the case for corpus analysis. If the grammar is productive, it should leave a record in the corpus through earlier adult productions, so situations like this should be quite rare.

Even if they do arise, they would do nothing more than add yet another argument for supplementing corpus analysis, which can only tell us what language producers did in the past, with experimental tests of contemporary speakers, which can tell us what language producers and comprehenders are doing today.

In summary, then, the analysis of corpora as evidence for underlying grammar is part of the standard methodology of generative phonology, since contrary to the anti-corpus rhetoric, frequency does correlate with grammaticality, and it is not a purely empiricist exercise because the researcher must start with prespecified hypotheses. In the remainder of this paper, we will see how these notions can be implemented quantitatively.

3. Quantitative evidence for grammar in phonological corpora

In this section I show how a hypothesized phonological grammar, represented in the familiar notation of Optimality Theory, can be tested against the statistical null hypothesis that the grammar does no work in accounting for the dictionary data. The mathematical background is given in section 3.1, where I review the notion of constraint weights and the statistical models designed to find them. In section 3.2 I go beyond the current literature by describing methods for testing the statistical significance both of individual OT constraints and of their ranking. The methods are illustrated in analyses of a pattern in the Formosan language Pazih.

3.1 Mathematical modeling of Optimality Theory

We want a model of phonological grammar that defines empirical success in terms of explicit prespecified components (as argued in section 2.3), and it would be even better if the model were mathematically simple and already familiar to practicing phonologists. Fortunately, such a model exists: Optimality Theory. OT, the lingua franca of contemporary theoretical phonology, claims that a grammar consists of the strict ranking of a set of universal constraints. This not only defines empirical success in an explicit way (a constraint either plays a statistically significant role in accounting for the data, or not, and likewise for a ranking), but the notions of constraints and constraint ranking turn out to provide a convenient framework for statistical formalism.

I start in 3.1.1 with background on the implicit role of constraint weights in Optimality Theory, then review the automatic fitting of weights in 3.1.2.

3.1.1 Optimality Theory and harmony theory

The idea underlying the methods described below is the relationship between OT and harmony theory (Prince and Smolensky 1993/2004, 1997). Since this relationship may not be familiar to all readers, I first review it here.

Both OT and harmony theory assume that knowledge (e.g. of a grammar) can be described in terms of violable constraints. The major difference is that unlike harmony theory, competition between constraints in OT is resolved via strict ranking: if constraint Con1 outranks constraint Con2, then Con2 can only choose the output if Con1 happens not to discriminate between the candidate outputs. This logic is familiarly expressed in so-called tableaux like those in (1), where the stars represent violations by the given candidate outputs of the given constraints, ranked by the grammar (here Con1 >> Con2). In each tableau, the pointing finger indicates the candidate output that is in the set of candidates that least violate the highest-ranked constraint, and among those, is also in the set of candidates that least violate the next-highest-ranked constraint, and so on.

(1) a.

InA	Con1	Con2
☞ OutA1		*
OutA2	*	

b.

InB	Con1	Con2
OutB1		*
☞ OutB2		

c.

InC	Con1	Con2
OutC1	*	*
☞ OutC2	*	

There is a mnemonic for teaching the logic of OT that is also quite useful in explaining its relation to harmony theory. Namely, one imagines that the blank cells in the tableau are zeroes, and the stars are ones (or whatever the number of stars is). Then one reads the rows of digits, from left to right, as comprising a number in the ordinary decimal system (as long as no cell has more than nine stars). The optimal candidate will then be the one associated with the lowest valued number. In the case of the tableaux in (1), this procedure yields the values and winning

candidates shown in (2).

(2) a. OutA1: 01 = 1
OutA2: 10 = 10 1 < 10, so OutA1 wins

b. OutB1: 01 = 1
OutB2: 00 = 0 0 < 1, so OutB2 wins

c. OutC1: 11 = 11
OutC2: 10 = 10 10 < 11, so OutC2 wins

This procedure works because a value notated in the decimal system represents the sum of the digits weighted so that the further left a digit is, the heavier its weight. Thus "24" represents 2 times ten plus 4 times one; the leftmost digit is associated with a higher weight (ten) than the rightmost digit (one). Like the decimal system, OT ranking is a kind of weighting system that gives higher weights to the higher-ranked constraints. Also like the decimal system, OT ranking is designed so that no lower-ranked constraint, or gang of constraints, can "outvote" a higher-ranked constraint. The decimal system does this by using digits ("violations") that never go over nine, and the implicit weights in OT have the same effect.

The logic behind (2) can be expressed more formally as in (3a), where $Star_i$ represents the number of violations of constraint Con_i and w_i represents the weight of Con_i. The systematic increase in constraint weight that forces constraints to be strictly ranked is ensured by the conditions in (3b).

(3) a. Candidate harmony value = $w_1 \times Star_1 + w_2 \times Star_2 + ... + w_n \times Star_n$
b. $w_1 = base^{n-1}$, $w_2 = base^{n-2}$, ... $w_n = base^0$, and base > max(Star)

Harmony theory is a generalization of OT (or, historically speaking, OT is a special case of harmony theory), where there are no built-in restrictions on the weighting system. That is, it assumes (3a) but not (3b). Thus it is possible for a lower-weighted constraint to "outvote" a higher-weighted one, or for a set of lower-weighted constraints to "gang up" on higher-weighted ones.

Although harmony theory predates OT, there has recently been a resurgence of interest in it (and variations on it), for two major reasons. First, some argue that its "fuzzy" ranking of constraints does a better job act capturing actual language data (e.g. Keller 2000, Pater et al. 2007b). Second, some are attracted by its mathematical connection to well-established

techniques in computational linguistics (e.g. Pater et al. 2007a; Hayes and Wilson to appear; Goldwater & Johnson 2003). The latter reason is more relevant to the goals of this chapter, since the automatic setting of constraint weights from corpus data underlies the statistical tests of OT grammar that I propose later.

3.1.2 Loglinear models

With corpus data, the most useful type of weight-fitting model is the family of so-called loglinear models. In this section I unpack the term "loglinear" into pieces, starting with "linear."

Loglinear models are a kind of linear regression model. The goal of linear regression is to analyze a scatterplot of data, where each dot in the scatterplot represents a pair of observed input and output values, in order to find the best linear fit (straight line) through the cloud of dots. This line will then be our best available description of the relationship between the input and output. The more linear the cloud of dots, the better the line describes the actual data. The associated *p* value then indicates that how probable it is that the cloud is linear purely by accident.

If the best-fitting line slopes upward (a positive correlation), this means that increasing input values are associated with increasing output values. Similarly, if the line slopes downward, the correlation is negative. Algebraically, the slope is expressed as a weight on the input variable, so we end up with the equation for the best-fit line shown in (4) ("Intercept" represents the value of the output when the input value is zero, where the line intercepts the y-axis).

(4) Output = Intercept + Weight × Input

The weight and intercept are not chosen by hand, but are calculated automatically from the data by a method called maximum likelihood estimation. This method finds the parameter values for the line maximizing the likelihood that this line underlies the cloud of data.

A similar procedure applies if we are trying to predict the output from two or more input values simultaneously. To take an example from psycholinguistics, the naming time (i.e. the time to initiate speech) for reading aloud a Chinese character depends, among other things, on how many strokes it has (affecting the time needed for visual processing) and on how many phonemes it has (affecting the time needed for phonological processing). Very likely the effects of strokes and phonemes on naming time don't interact with each other, so naming time will be related to the sum of the two effects (see Sternberg 1998 for discussion of real examples of this kind of analysis). Naming time is presumably also influenced by other factors, unknown or even random. A three-dimensional scatterplot of this situation would be a boxlike structure, with the base plane defined by two axes representing, respectively, the number of strokes and the number of

phonemes, and elevation off the base representing naming time. Due to the randomness, inside the box we will see a cloud of data points rather than a nice neat line. The line that best fits this cloud will have a slope that can be described separately relative to the Strokes and Phonemes axes. That is, the orthogonal axes decompose the line's slope into two mutually independent components (imagine shadows projected on the walls of the box). Algebraically, the situation can be expressed with a regression equation like that in (5).

(5) Output = Intercept + Weight1 × Input1 + Weight2 × Input2

In the case of testing an OT grammar, the input variables are the numbers of violations of our proposed constraints, and the weights are the constraint weights. To explain what the output represents, we must now unpack the "log" part of "loglinear."

It's not really appropriate to treat type frequencies in a corpus the same way as a continuous-valued measure like naming time. Type frequencies can be thought of as category sizes or probabilities (e.g. the probability of a random corpus item falling into one category rather than another). These measures relate to discrete observations (i.e. the individual corpus items). If we simply add up the factors affecting these category sizes or probabilities, as in ordinary linear regression, we won't describe the output correctly. For one thing, neither category sizes nor probabilities can go below zero, yet the right side of the equation describes an infinitely long line that goes below zero. Another problem is that independent influences on probability shouldn't be added, but multiplied (e.g. in a dice game the probability of rolling a six is 1/6, and the probability of rolling two sixes is (1/6) × (1/6)). Putting these points together, we really should be looking at an equation with a product on the right side, not a sum. Unfortunately, products are harder to deal with mathematically than sums.

The solution is to exploit the mathematical trick shown in (6), which may vaguely ring a bell for those with memories stretching back to algebra or calculus classes. Namely, a product containing two variables x and y can be converted into the sum of x and y if we start out with x and y as powers of some base value (here, the famous constant e = 2.71828+) and then take the logarithm of the same base. Hence if we define our regression equation carefully, we can take the logarithm of both sides and convert the right side to a conveniently linear sum (and the left side to a value that can, in principle, range infinitely below and above zero). This is a loglinear model.

(6) $\log_e[(e^x)(e^y)] = \log_e[e^{x+y}] = x + y$

Loglinear models are all closely related to each other mathematically, but different variants have different strengths. For most linguists, the most familiar type of loglinear model is logistic

regression, the workhorse at the heart of the VARBRUL program widely used in sociolinguistics (Mendoza-Denton et al. 2003). Logistic regression is used when observations are binary, such the choice among two competing allomorphs in a sociolinguistic corpus. This type of regression doesn't suit our purposes, however, since we are interested in the sizes of the categories attested in a corpus, not in competing variants.

Another type of loglinear model is more familiar to computational linguists, who express maximum likelihood in terms of the related information-theoretic notion of maximum entropy. Recently there have been a growing number of studies applying maximum entropy models to harmony theory, including Goldwater and Johnson (2003) and Hayes and Wilson (to appear), with Pater et al. (2007a) using a different algorithm (linear programming) for a similar purpose (learning harmonic grammars). Like non-loglinear OT learning algorithms (e.g. Tesar and Smolensky 1998, Boersma and Hayes 2001), a maximum entropy model is given a set of harmonic constraints, a set of input forms, a set of candidate outputs for each input, and an indication about which of these candidates is the winning output. The models then learn the best weights consistent with the trained input-output pairs, if there are any.

These types of loglinear models are automatic grammar learners, and unlike the logistic regression models used by sociolinguists, are not intended to test for statistical significance. This may be partly because scientific hypothesis testing is not the traditional focus of the engineering-oriented maximum entropy literature, but perhaps also because the modelers recognize the possibility, noted at the end of section 2.3, that actual language learners may not care about statistical significance. These models also do not test for the statistical reliability of constraint ranking, which makes sense given that the algorithms are designed to learn harmonic grammars, not strictly-ranked OT grammars.

What kind of loglinear regression model would be most appropriate for testing OT hypotheses about corpus data? Like logistic regression, the left side of the equation should represent corpus observations and the right side our grammatical hypotheses about them. Unlike logistic regression, however, the left side of the equation should relate to the sizes of categories, namely the type frequencies for structures predicted to be grammatical vs. structures predicted to be ungrammatical.

Fortunately, there is a well-established type of loglinear model ideal for our purposes. It is called Poissonregression, named after the French mathematician Siméon-Denis Poisson (e.g. Agresti 2002). Chance probability in this type of regression is defined in terms of the Poisson distributions associated with count data. Poisson distributions are asymmetrical, since counts cannot go below zero and small values are more common than large ones (counting up to a higher number entails counting up to lower ones first, but the reverse is not the case).

In the following section, then, I show how Poisson regression can be applied to test the

statistical significance of OT constraints and their ranking.

3.2 Testing OT grammars

In this section I show how OT-based grammatical hypotheses can be tested statistically, giving all of the commands necessary for running the analyses using R, the free statistics software package (R Development Core Team 2007; see also Baayen to appear, Hammond this volume). The procedures are illustrated with data from the Formosan language Pazih (or Pazeh; data from Li and Tsuchida 2001). Other aspects of the data analysis problems posed by the Pazih data are discussed in Myers (to appear). Pazih represents a good test case of corpus analysis methodology because like a growing number of languages studied by phonologists, Pazih is nearly extinct, and very soon all the world will have left is a fixed record of the language as it was once spoken.

The logic works roughly as follows. Since the mathematics of linear modeling (loglinear or otherwise) treats components on the right side of the equation as orthogonal to each other, we should be able to test the independent contributions of each constraint; we may find, for example, that one constraint is statistically significant but another is not. To test a ranking hypothesis, we can compare the constraint weights; the constraint ranking Con1 >> Con2 would be supported if we found that the weight w1 was significantly larger than the weight w2. Testing individual constraints is a more basic task, so I describe this first, in section 3.2.1. The test for ranking is discussed in section 3.2.2.

3.2.1 Testing OT constraints

Pazih has a set of 45 morphemes consisting of reduplicated CVC syllables with an epenthetic medial vowel. In 33 of these morphemes, the epenthetic vowel is identical to that of the reduplicated syllables, as in (7). However, in 12 morphemes, the epenthetic vowels do not show vowel harmony, as in (8) (note the minimal pair in (7b) vs. (8b)). It is this small corpus of 45 items that is the focus of the next few sections.

(7) a. bak-a-bak "native cloth"
b. hur-u-hur "steam, vapor"

(8) a. bar-e-bar "flag"
b. hur-a-hur "bald"

Setting aside a number of interesting issues (see Myers to appear), one way to analyze this pattern would be to describe regular words like those in (7) as having no underlying value for the epenthetic vowel, which receives its surface form by obeying a universal vowel harmony constraint AgreeV. Exceptions like those in (8), where the value of the epenthetic vowel appears to be unpredictable, specify this value underlyingly, and preserve it via a faithfulness constraint IdentV. To prevent AgreeV from nullifying IdentV in (8), we can hypothesize the ranking IdentV >> AgreeV. This analysis can be summarized in the tableaux (9) and (10) for regular words and exceptions, respectively.

(9) "steam"

hur-V-hur	IdentV	AgreeV
☞ hur-u-hur		
hur-a-hur		*

(10) "bald"

hur-a-hur	IdentV	AgreeV
hur-u-hur	*	
☞ hur-a-hur		*

Note that implicit quantification plays a crucial role in this argument, allowing us to classify the words in (7) as regular and those in (8) as exceptions. If it had been the case that the great majority of words had epenthetic vowels different from the base syllables, we would never have considered any role for AgreeV in this language, with apparently "harmonizing" examples dismissed as statistical flukes.

Now we want to put this implicitly quantitative argument on firmer statistical footing. Namely, are 33 vowel-harmonizing morphemes out of 45 truly enough to justify AgreeV? After all, 12/45 exceptions is 27%, quite a substantial proportion, especially given that Pazih only has four phonemic vowels to work with. Justifying IdentV seems even more difficult, given the argument in section 2.2 that lexical exceptions should be considered ungrammatical. For example, if there were 44 harmonizing morphemes and only one exception, the proportion of data points in support of IdentV would drop to 1/45, which seems to be too low to justify its existence in the grammar, given the mind's extra-grammatical ability to memorize arbitrary exceptions.

Poisson regression can help clarify the situation. The first step is to classify words in the corpus in terms of the grammar. As in the schematic example in 2.2., we can think of each proposed constraint as dividing the corpus into two categories, grammatical vs. ungrammatical

relative to this constraint. In the case of the Pazih data, the hypothesized grammar puts lexical items into four categories, defined by the two evaluations associated with AgreeV (violate vs. obey) times the two evaluations associated with IdentV (violate vs. obey). By crossing the two categorization schemes, we define orthogonal axes that decompose their contributions independently, allowing us to test both constraints at the same time in the same statistical model.

To make the constraint evaluations consistent with the procedure described in section 3.1.1, we code violations with 1 (star) and use 0 (blank cell) for instances where the constraint is obeyed. We then count the number of items of each type in the corpus, and tabulate the information as in (11).

(11)

Count	AgreeV	IdentV
33	0	0
0	0	1
12	1	0
0	1	1

The table in (11) should not be confused with an OT tableau. First, it summarizes information about many items, not just a single item. Second, it does not represent input-output pairs. Rather, it encodes item representations in terms of the constraints they violate. In doing this, it is similar to the OT-based representational scheme proposed in Golston (1996), except that it also permits faithfulness constraints like IdentV to be included as part of the description. Thus the descriptions encode both output forms (via markedness constraints) and input forms (via faithfulness constraints). From the very start, then, the data are encoded in a theory-governed way. As argued in section 2.3, this is necessary for hypothesis testing in corpus analysis to avoid circularity.

Now we want to use Poisson regression to fit an equation like that in (12). Since the values of AgreeV and IdentV represent violations, we expect that the weights for both will be negative, since counts should be higher for categories where constraints are obeyed (evaluated as 0) and lower for categories where constraints are violated (evaluated as 1). In addition, we hope that both weights are significantly different from zero. That is, the chance probability of getting weights with magnitudes as great as ours should be "sufficiently low" (conventionally, $p < 0.05$ is considered to be statistically significant).

(12) $\log_e(\text{mean}_{\text{Poisson}}) = \text{Intercept} + \text{Weight1} \times \text{AgreeV} + \text{Weight2} \times \text{IdentV}$

Unfortunately, before continuing this example we must make an adjustment to accommodate an intrinsic limitation of the maximum likelihood estimation method. Namely, this method assumes that none of the correlations are perfect, so if any is, the algorithm will crash and the results will be nonsense. In our case, Count = 0 always occurs whenever IdentV = 1. The effect is that IdentV = 0 holds for all attested tokens, which makes phonological sense (vowel harmony presumably doesn't change underlying feature values), but it wreaks mathematical havoc. There are alternative methods that can avoid this limitation; one promising one is exact loglinear regression, which has the added advantage of giving equally reliable results for any sample size (e.g. Agresti 2002). Unfortunately it is computationally intensive, since *p* values are computed by counting all logically possible outcomes rather than relying on distribution-based estimates, and R currently does not implement the necessary algorithms.

One simple (and perhaps simple-minded) way out is to tweak the data to make the correlations less than perfect. This can be done by replacing all zero counts with one, as if all categories were non-empty, yielding the modified data table in (13). This would entail the appearance in the corpus of two theoretically anomalous items, one obeying AgreeV but violating IdentV (e.g. /hur-a-hur/ > [hur-u-hur]) and one violating both (e.g. /hur-a-hur/ > [hur-i-hur]). Fortunately, besides dealing with the mathematical glitch, we can justify their inclusion for another pragmatic reason: including such fake items slightly weakens the evidence for both AgreeV and IdentV, so if they still manage to be statistically significant, our confidence in them should be even stronger.

(13) Contents of Pazih.txt

Count	AgreeV	IdentV
33	0	0
1	0	1
12	1	0
1	1	1

Now we are ready to run the Poisson regression. To do this in R, we must save the table in (13) as a tab-delimited text file. The easiest way to do this is by creating the table in a spreadsheet program like Excel, including the headings, and then copying the cells and pasting them into a text editing program (like Window's Notepad). Let's call this file Pazih.txt.

After R has been downloaded, installed, and started up, we change its file directory to the location of the data file so that R can find it. R is a command-line program, so we must now type

in commands in the main R window to load the data and name it, as in (14) (note that file names must be given inside quotes, since they are treated as text strings). For our file Pazih.txt, the command would be as follows. In these examples, underlined elements represent names that should be changed depending on your particular circumstances, while the rest should be entered exactly as shown. Note that <- is a leftward pointing arrow, assigning a value to a variable, and text following # is ignored by R, so we can add explanatory comments for us humans. The *F(X)* syntax represents a function F operating on an argument (or set of arguments) *X*.

(14) Tableau <- read.table("Pazih.txt",T) # Load data with headers
attach(Tableau) # Make column names available

Once this is done, running the Poisson regression simply involves typing the command in (15). The *Y ~ X1 + X2* syntax represents the regression equation, the *glm* command runs a generalized linear model (i.e. a regression model where the linear part is created by transformation, in this case by taking the logarithm), the *family* tag indicates that we want to use Poisson distributions for count data, the *summary* command generates a compact overview of the results, and the *$coefficients* tag pulls out from this summary just the table showing the constraint weights and their associated *p* values.

(15) summary(glm(Count ~ AgreeV + IdentV, family = poisson))$coefficients

In the case of the data in Pazih.txt, the command in (15) gives the table in (16).

(16)

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	3.482875	0.174234	19.98968	6.77E-89
AgreeV	-0.96141	0.32609	-2.9483	3.20E-03
IdentV	-3.11352	0.722643	-4.30851	1.64E-05

The estimates are the constraint weights and *Pr(>|z|)* represents the *p* values (the standard errors and *z* values are used to compute the *p* values). Note first that the weights for AgreeV and IdentV are both negative, as desired: violations of them (coded 1) are associated with lower type frequencies in the corpus. Moreover, since "*x*E-*y*" in the *p* value column means $x/10^y$, the *p* values for both constraints represent statistically significant results (AgreeV: $p = 0.0032 < 0.05$; IdentV: $p = 0.0000164 < 0.05$). This is so despite the need to add spurious data points that weakened evidence for the two constraints.

In contrast with this procedure, OT and harmonic grammar models that automatically learn grammars from corpus data only assign weights, but do not test if they are statistically significant. For example, with the (original unaltered) Pazih data, the Harmonic Grammar with Linear Programming (HaLP) model of Pater et al. (2007a) assigns a weight of 2.0 to IdentV and a weight of 1.0 to AgreeV, but it doesn't give any associated *p* values. Moreover, these same weights will be derived even if only one of the 45 items is non-harmonizing (giving almost no evidence for IdentV), or if all but one of the 45 items is non-harmonizing (giving almost no evidence for AgreeV). Other learning algorithms, such as the Gradient Learning Algorithm of Boersma and Hayes (2001), as implemented in Praat (Boersma and Weenink 2007) share this behavior. Whether or not actual children show one-trial learning of grammars, such models are certainly not helpful for researchers interested in what a corpus reveals about the grammar underlying it. Since a corpus is performance data, it is inherently noisy. Thus a complete analysis must include a measure of this noisiness, which, roughly speaking, is what the *p* values represent.

Because of the way regression factors out the contributions of individual input variables, the procedure described here can be generalized to any number of constraints. For example, with three constraints, the Count column would represent eight (2^3) type frequencies. The procedure is a bit more complex if the analysis includes constraints that evaluate some items with more than one star, such as inherently gradient constraints or categorical constraints that happen to be violated more than once by the same item. The difficulty is not gradience itself, since for any given constraint the regression approach can test whether the number of stars and the associated category sizes are inversely correlated. However, multiply violated constraints will increase the number of categories that must be tested. For example, an analysis with two constraints, each of which can give evaluations of up to two stars, defines nine (3^2) categories, in contrast to the four categories of the two-constraint analysis of Pazih. If constraint evaluations can be continuous, as suggested in Kirchner (1997), the procedure breaks down completely.

Note that so far, the procedure is actually consistent with harmonic grammar as well as OT, since both assume constraint weighting. It should also be noted that even if a constraint proves to be unsupported statistically (empirically), there may be a priori (rationalist) reasons for maintaining it in the analysis anyway. In OT, for example, markedness constraints that never apply in some language because they are dominated by faithfulness constraints are assumed, nevertheless, to exist, since it keeps the overall (universal) theory simpler (i.e. grammars can be described as varying only in constraint ranking, but not in constraint inventory). Another way to put it is that absence of evidence is not the same as evidence of absence.

3.2.2 Testing OT constraint ranking

The observant reader will have noticed that not only are the weights in (16) for AgreeV and IdentV negative and statistically significant, as desired, but the magnitude of the weight for IdentV (-3.11) is greater than that for AgreeV (-0.96). This is also consistent with the OT analysis, since we expect that AgreeV, though active in the grammar, is nevertheless ranked lower than IdentV. With only two constraints Con1 and Con2, each of which is violated no more than once, finding that the difference in the associated weights is in the appropriate direction (w1 > w2) is sufficient to demonstrate the strict OT ranking of Con1 >> Con2. This is demonstrated in (17) and (18) using the original OT tableaux for Pazih with the regression-determined weights.

(17) "steam"

a.

hur-V-hur	IdentV w = -3.11	AgreeV w = -0.96
☞ hur-u-hur		
hur-a-hur		*

b. hur-u-hur: (-3.11)(0) + (-0.96)(0) = 0
hur-a-hur: (-3.11)(0) + (-0.96)(1) = -0.96 |0| < |-0.96|, so hur-u-hur wins

(18) "bald"

a.

hur-a-hur	IdentV w = -3.11	AgreeV w = -0.96
hur-u-hur	*	
☞ hur-a-hur		*

b. hur-u-hur: (-3.11)(1) + (-0.96)(0) = -3.11
hur-a-hur: (-3.11)(0) + (-0.96)(1) = -0.96 |-0.96| < |-3.11|, so hur-a-hur wins

Yet how can we be sure that this difference in constraint weights is not a statistical fluke? Fortunately, there turns out to be a very simple way to find out (as pointed out to me by James S. Adelman, personal communication, September 7, 2007). The technique builds on the fact that linear regression equations involve sums of components on the right side. Given a pair of equations that are identical except for an extra component in one of them, we can conclude that the extra component makes a significant contribution if the more complex equation fits the

observed data better. In the case of loglinear regression, the relevant test for comparing simple and complex equations is called the analysis of deviance (analogous to the more familiar analysis of variance or ANOVA, used when comparing ordinary linear regression equations).

To see how this leads to a test for comparing weights, let's start with the two equations in (19). By comparing them, we can test whether InputY makes a significant contribution to explaining Output.

(19) a. Output = Intercept + Weight × InputX
b. Output = Intercept + Weight1 × InputX + Weight2 × InputY

Now suppose that InputX = Input1 + Input2 and InputY = Input1 - Input2. This gives the equivalent equations in (20).

(20) a. Output = Intercept + Weight × (Input1 + Input2)
b. Output = Intercept + Weight1 × (Input1 + Input2) + Weight2 × (Input1 - Input2)

Next we perform a bit of algebra. First we transform (20a) as in (21a), showing that this equation assumes that the weights for both inputs are identical. Then we transform (20b) as in (21b), which reorganizes the components around the two inputs rather than around the weights.

(21) a. Output = Intercept + Weight1 × Input1 + Weight2 × Input2,
where Weight1 = Weight2

b. Output = Intercept + Weight1 × (Input1 + Input2) + Weight2 × (Input1 - Input2)
= Intercept +
(Weight1 × Input1 + Weight1 × Input2) + (Weight2 × Input1 - Weight2 × Input2)
= Intercept +
(Weight1 × Input1 + Weight2 × Input1) + (Weight1 × Input2 - Weight2 × Input2)
= Intercept + (Weight1 + Weight2) × Input1 + (Weight1 - Weight2) × Input2

Since the values of Intercept, Weight1, and Weight2 are set (by maximum likelihood estimation) separately for each of the two equations, the equation in (21b) is formally equivalent to that in (22b), which is simply our usual two-constraint model that we saw earlier in (5).

(22) a. Output = Intercept + Weight1 × Input1 + Weight2 × Input2 [Weight1 = Weight2]
b. Output = Intercept + Weight1 × Input1 + Weight2 × Input2

The end result is that we now know that the model in (22b), which assumes different weights for the two constraints, is actually just an additive extension of the model in (22a), which assumes identical weights for both constraints. Hence it is legitimate to use a regression equation comparison method like analysis of deviance to compare the two models, which, in effect, tests the hypothesis that the weights are different against the null hypothesis that they are the same.

In the case of the Pazih data, the two models can be expressed in R as shown in (23). The model in (23a) assumes identical weights, and the model in (23b) assumes different weights. By default, R interprets arithmetic expressions in regression models as abbreviations for equations like that in (22), so in order to express ordinary addition as in (20a), we must embed the addition symbol inside the *I* ("as is") function in the command in (23b).

(23) a. NoRanking <- glm(Count ~ I(AgreeV + IdentV), family = poisson)
b. Ranking <- glm(Count ~ AgreeV + IdentV, family = poisson)

If we like, we can see the outcome of these two models with the *summary* function and *$coefficients* tag as before, as in (24).

(24) a. summary(NoRanking)$coefficients
b. summary(Ranking)$coefficients # Same outcome as in (16)

However, here the real goal is to compare the two models. This is done using the command in (25), where "Chisq" represents the chi-square test (which may sound familiar, given that versions of it are often taught in introductory statistics classes).

(24) anova(NoRanking, Ranking, test = "Chisq")

The output of this command, shown in (25), indicates that the more complex model has a much smaller residual deviance, meaning that it fits the data better. This improvement is statistically significant, as shown by the chi-squared $p = 0.0012 < 0.05$.

(25) Model 1: Count ~ I(AgreeV + IdentV)
Model 2: Count ~ AgreeV + IdentV

	Resid. Df	Resid. Dev	Df	Deviance	P(>|Chi|)
1	2	11.0185			
2	1	0.4677	1	10.5508	0.0012

Hence we can conclude that not only are both IdentV and AgreeV individually supported by the corpus data, but their hypothesized ranking is also supported. More precisely, this test shows that the two constraint weights are significantly different. Since we already know that the absolute value of IdentV's weight is greater than that of AgreeV's weight, this significant difference is consistent with the claim that IdentV >> AgreeV.

A bit more algebra is necessary if the hypothesized grammar involves constraints that can give evaluations of more than one star. In this case, the requirement that Weight1 be larger than Weight2 is not sufficient, since we don't want a strong violation of a lower-ranked Con2 (many stars) to override the opinion of a higher-ranked Con1, even if this opinion is expressed less strongly (e.g. with one star). Thus to demonstrate Con1 >> Con2, we actually have to demonstrate the relationship shown in (26a), which is equivalent to (26b). In other words, we have to falsify the null hypothesis in (26c).

(26) a. Weight1 × max(Star1) > Weight2 × max(Star2)
b. Weight1 > Weight2 × [max(Star2)/max(Star1)]
c. Weight1 = Weight2 × [max(Star2)/max(Star1)]

Since it is defined by the hypothesized grammar and data set as a whole, not by individual corpus items, max(Star2)/max(Star1) is a constant; let's call it MaxDif. A bit of algebra then results in the no-ranking hypothesis being expressed in R as in (27a), with its competitor as in (27b).

(27) a. NoRanking <- glm(Count ~ I(MaxDif × Con1 + Con2), family = poisson)
b. Ranking <- glm(Count ~ Con1 + Con2, family = poisson)

Testing ranking hierarchies with more than two constraints requires still more work. First, we need more algebra. Take the hypothesized constraint ranking Con1 >> Con2 >> Con3. Among other things, this implies that the lower-ranked constraints Con2 and Con3 cannot "gang up" and collectively override Con1. In terms of constraint weights, the equation in (28a) must hold. That is, we want to falsify the null hypothesis in (28b) (returning, for simplicity, to the case where no constraint can assign more than one star).

(28) a. Weight1 > Weight2 + Weight3
b. Weight1 = Weight2 + Weight3

After some algebraic manipulation, the no-ranking and ranking models would be expressed in R as in (29a) and (29b), respectively.

(29) a. NoRanking <- glm(Count ~ I(Con1 + Con2) + I(Con1 + Con3), family = poisson)
b. Ranking <- glm(Count ~ Con1 + Con2 + Con3, family = poisson)

The second complexity is that the ranking Con1 >> Con2 >> Con3 implies more than one ranking claim, namely both Con1 >> {Con2, Con3} and Con2 >> Con3. Both concern the same data set, yet unlike the simultaneous tests of constraint weights within a single regression equation, these tests are not conditional on (orthogonal to) each other. Instead, each is conducted as if the other test hadn't been done at all. Without some adjustment, this is cheating. Roughly speaking, it's as if we are asking two related questions without taking into account that the answer to one may affect the answer to the other. The more chances we have to guess the answer, the easier it is for at least one of the answers to be "right" by accident. A crude but simple approach to the problem would be to penalize ourselves more the more tests we run, using so-called Bonferroni adjustment. This involves dividing the critical value demarcating significance by the number of tests. With three constraints, there are two tests, so for each test we must reach $p < 0.05/2 = 0.025$. Bonferroni adjustment is hardly an ideal technique (Perneger 1998) but it's conceptually simple and widely used.

It is reasonable to ask whether the procedure proposed here for comparing constraint weights is really testing constraint ranking in standard OT. The simple answer is no, not exactly. This is because there is no single set of "true weights" in an OT analysis. To go back to the mnemonic introduced in section 3.1.1, instead of using the decimal system we could have used binary or a system with different bases for each constraint. Indeed, the many competing OT and harmonic grammar algorithms all produce different constraint weights from the same data. This calls into question the assumption that a significant difference in Poisson regression weights necessarily supports ranking in the traditional OT sense.

Nevertheless, the proposed procedure has an intuitive plausibility. A Poisson regression test of individual constraints is, in a sense, a formalization of the informal analysis performed by phonologists when they check to see how well patterns generalize across a corpus. Moreover, ranking and data coverage are related, even in the traditional methodology. For example, an undominated constraint not only has a ranking status (at the top) but it also provides a true description of all of the corpus data. Similarly, a constraint with a slightly lower rank should be expected to be violated more often in the corpus than one with a much lower rank. In the present case, not only is IdentV never violated in the actual data while AgreeV is, but the evidence for IdentV itself is quite strong, given that 27% of the corpus items provide positive evidence for it.

By contrast, if only 5 out of 45 of the items required IdentV, the weight associated with it would be smaller, hence closer to that of AgreeV (though still larger than it). Indeed, when the proposed ranking test is applied to this alternative version of Pazih, the two weights are not significantly different, fitting with our intuitive expectations that such a corpus would not provide convincing evidence of their ranking, given the possibility of extra-grammatical exceptions.

4. Conclusions and future prospects

If I've convinced the reader of nothing else, I hope I've shown that it is hypocritical for generative linguists to criticize the use of corpus data, since generative phonologists actually depend on such data almost exclusively. Anti-corpus rhetoric actually boils down to the need for a priori assumptions in grammar learning, which I agree is necessary. Yet I've also argued that it is also legitimate to set aside the question of acquisition and focus on the corpus as a source of evidence about the adult (or diachronic) grammar that produced it. This is also the logic underlying the vast bulk of the literature in generative phonology, though it is not always expressed this way. I then made a case for considering corpus frequencies as informative about grammar. The usual retort that a corpus is mere performance data, affected by many things other than grammar, is a non sequitur given that all linguistic data represent performance, even acceptability judgments. Finally, I demonstrated a set of simple procedures for putting quantitative teeth into this view of corpus analysis, so that prespecified OT grammars, composed of constraints and their ranking, can be tested against dictionary data. It turns out that these procedures can be built entirely on well-established statistical methods, namely loglinear (poisson) regression equations and comparisons between them.

Much more could be said about all of this, but I will restrict myself to just three additional points. First, the quantitative procedures assume that dictionaries result solely from OT constraints and purely random noise. Since type frequencies are also affected by systematic extra-grammatical forces, the proposed methods may find statistically significant support for a hypothesized constraint when the data pattern actually results from borrowing or analogy. In principle, the solution to this challenge is simple. Namely, include additional input variables in the statistical model to represent these extra-grammatical forces. The mathematics of regression will then factor them out separately from the hypothesized grammatical constraints, so if the latter remain significant, they must play some genuinely predictive role after all. The technical details get a bit complex, however, so I will not discuss them here.

Second, not only can the method for testing individual constraints be applied to non-OT harmonic grammar, but it is even appropriate for testing the surface reliability of rules in a derivational framework. Rule ordering is not a mere historical curiosity, since standard OT is

incapable of handling opacity, and among the many solutions offered for the problem is the use of ordered constraints or constraint blocks (e.g. Kiparsky 2000, Rubach 2000). The loglinear method applies to rules because the statistics are merely testing whether each claim is supported by the corpus; finding the weight is merely a bonus. For example, if we expressed vowel harmony in Pazih in terms of an autosegmental rule, we could still count how often the rule is applied (33/45 items) and how often the rule is blocked (12/45 items). Moreover, there also happen to be statistical methods for testing rule ordering hypotheses from corpus data, developed just before the OT revolution (Sankoff and Rousseau 1989). It is not immediately clear how they could be incorporated into the present methodological framework, but the question seems worthy of exploration.

Finally, though I have tried to make the background and instructions for the statistical procedures as simple as possible, it is likely that they are still too technically intimidating for the average working phonologist. It would therefore be beneficial if they could be carried out by a software tool running behind a fully intuitive, non-threatening interface. I'm currently working on a program with this goal, called MiniCorp (to complement my judgment-analysis program MiniJudge; Myers 2007). Computationally minded readers are encouraged to compete with me to produce a better program.

References

Agresti, Alan. 2002. *Categorical data analysis*, 2nd ed. Hoboken, NJ: Wiley-Interscience.

Baayen, R. Harald. To appear. *Analyzing linguistic data: A practical introduction to statistics*. Cambridge, UK: Cambridge University Press.

Bailey, Todd M., and Ulrike Hahn. 2001. Determinants of wordlikeness: Phonotactics or lexical neighborhoods? *Journal of Memory & Language* 44:569-591.

Blevins, Juliette. 2004. *Evolutionary phonology: The emergence of sound patterns*. Cambridge, UK: Cambridge University Press.

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

Boersma, Paul, and David Weenink. 2007. Praat: doing phonetics by computer (Version 4.6.28) [Computer program]. Software package: http://www.praat.org/

Chomsky, Noam. 1957. *Syntactic structures*. The Hague: Mouton.

Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

Chomsky, Noam. 2002. *On nature and language*. Cambridge, UK: Cambridge University Press.

Chomsky, Noam, and Morris Halle. 1968. *The sound patterns of English.* Cambridge, MA: MIT Press.

Duanmu, San. 2004. A corpus study of Chinese regulated verse: phrasal stress and the analysis of variability. *Phonology* 21:43-89.

Frisch, Stefan A., and Bushra Adnan Zawaydeh. 2001. The psychological reality of OCP-Place in Arabic. *Language* 77.1:91-106.

Frisch, Stefan A., Janet B. Pierrehumbert, and Michael B. Broe. 2004. Similarity avoidance and the OCP. *Natural Language & Linguistic Theory* 22: 179-228.

Goldwater, Sharon, and Mark Johnson. 2003. Learning OT constraint rankings using a maximum entropy model. In J. Spenader, et al. (Eds.), *Proceedings of the Stockholm Workshop on Variation within Optimality Theory* (p. 111-120). Stockholm Univ.

Golston, Chris. 1996. Direct Optimality Theory: Representation as pure markedness. *Language* 72.4:713-748.

Hammond, Michael. 2007. Empirical methods in phonological research. In James Myers (Ed.) *In search of grammar: Empirical methods in the study of linguistic knowledge.*

Hayes, Bruce and Donca Steriade. 2004. Introduction: The phonetic bases of phonological markedness. In Bruce Hayes, Robert Kirchner, and Donca Steriade (Eds.) *Phonetically based phonology* (pp. 1-33). Cambridge, UK: Cambridge University Press.

Hayes, Bruce, and Colin Wilson. To appear. A maximum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry*.

Inkelas, Sharon, C. Orhan Orgun, and Cheryl Zoll. 1997. The implications of lexical exceptions for the nature of grammar. In Iggy Roca (Ed.) *Derivations and constraints in phonology* (pp. 393-418). Oxford: Clarendon Press.

Keller, Frank. 2000. *Gradience in grammar: Experimental and computational aspects of degrees of grammaticality*. PhD dissertation, University of Edinburgh.

Kenyon, John Samuel, and Thomas A. Knott. 1944. *A pronouncing dictionary of American English*. Springfield, MA: Merriam.

Kiparsky, Paul. 2000. Opacity and cyclicity. *The Linguistic Review* 17: 351-67.

Kiparsky, Paul. 2006. Amphichronic linguistics vs. Evolutionary Phonology. *Theoretical Linguistics* 32:217-236.

Kirchner, Robert. 1997. Contrastiveness and faithfulness. *Phonology* 14:83-111.

Li, Paul Jen-kuei, and Shigeru Tsuchida. 2001. *Pazih dictionary*. Taipei: Institute of Linguistics.

Mendoza- Denton, Norma, Jennifer Hay, and Stefanie Jannedy. 2003. Probabilistic sociolinguistics: Beyond variable rules. In Rens Bod, Jennifer Hay and Stefanie Jannedy (Eds.) *Probabilistic linguistics* (pp. 97-138). Cambridge, MA: MIT Press.

Myers, James. 2007. MiniJudge: Software for small-scale experimental syntax. *International Journal of Computational Linguistics and Chinese Language Processing* 12 (2):175-194.

Myers, James (to appear). Linking data to grammar in phonology: Two case studies. *Concentric*.

Ohala, John J. 1986. Consumer's guide to evidence in phonology. *Phonology Yearbook* 3: 3-26.

Pater, Joe, Christopher Potts, and Rajesh Bhatt. 2007a. Harmonic grammar with linear programming. Ms, University of Massachusetts, Amherst. ROA 872-1006. Software package: http://web.linguist.umass.edu/~halp/.

Pater, Joe, Rajesh Bhatt and Christopher Potts. 2007b. Linguistic Optimization. Ms, University of Massachusetts, Amherst.

Pater, Joe. To appear. The locus of exceptionality: Morpheme-specific phonology as constraint indexation. In Steve Parker (Ed.) Phonological argumentation. Equinox.

Perneger, Thomas V. 1998. What's wrong with Bonferroni adjustments. *British Medical Journal* 316 (7139): 1236-1238.

Port, Robert and Adam Leary. 2005. Against formal phonology. *Language* 85:927-964.

Prince, Alan, and Paul Smolensky. 1993. *Optimality Theory: Constraint interaction in generative grammar*. Published 2004, Blackwell.

Prince, Alan, and Paul Smolensky. 1997. Optimality: from neural networks to universal grammar. *Science* 275:1604-1610.

R Development Core Team. 2007. R: A language and environment for statistical computing [Computer software]. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. http://www.R-project.org.

Rubach, Jerzy. 2000. Glide and glottal stop in Slavic languages: A DOT analysis. *Linguistic Inquiry* 31:271-317.

Sankoff, David, and Pascale Rousseau. 1989. Statistical evidence for rule ordering. *Language Variation and Change* 1:1-18.

Sternberg, Saul. 1998. Discovering mental processing stages: The method of additive factors. In Don Scarborough and Saul Sternberg (Eds.) *An invitation to cognitive science, vol. 4: Methods, models, and conceptual issues* (pp. 703-863). Cambridge, MA: MIT Press.

Tesar, Bruce, and Smolensky, Paul. 1998. The learnability of Optimality Theory. *Linguistic Inquiry* 29:229-268.

Uffmann, Christian. 2006. Epenthetic vowel quality in loanwords: Empirical and formal issues. *Lingua* 116:1079-1111.