A Guide to Analysis in MaxEnt Optimality Theory

partial book draft, July 2026

To appear, after completion and final review, with Cambridge University Press

This distribution has Chapters 1-5 of nine total.

Bruce Hayes and Claire Moore-Cantwell

Department of Linguistics

UCLA

Contents

Chapter 1: Purpose and orientation **4**

1.1 Rationale for this book 4
1.2 MaxEnt as tool vs. theory 4
1.3 Why is it called "MaxEnt"? 5
1.4 Content 5
1.5 Our MaxEnt spreadsheets and their locations (Supplementary Materials) 6
1.6 Policy on mathematical presentation 7
1.7 Acknowledgments 7

Chapter 2: The analysis of variation in outputs **8**

2.1 Analyzing the K1 language in Classical OT 8
2.2 K2 and free variation 12
2.3 The interpretation of free ranking as probability ~~with partially ordered constraints~~ 14
2.4 K3: modeling arbitrary probabilities 15
2.5 MaxEnt: an introduction 16
2.6 K4: Perturbers as "hidden" phonology in variation 22
2.7 K5: Multiple perturbers 31
2.8 Visualizing the K5 pattern with shifted sigmoids 36
2.9 Very high weights and the approximation of zero 42
2.10 Some Excel tips 43
2.11 More MaxEnt software 46
2.12 Where to examine published analyses using MaxEnt 47

Chapter 3: Using MaxEnt to study lexical frequency-matching **48**

3.1 Tagalog nasal substitution 48
3.2 An initial MaxEnt analysis 51
3.3 Shifted sigmoids for Tagalog 57
3.4 Frequency-matching grammars and lexical listing 59
3.5 Frequency-matching more broadly 60

Chapter 4: The MaxEnt Math **63**

4.1 Studying the MaxEnt formula 63
4.2 Relationship of MaxEnt to classical Optimality Theory 67
4.3 Relationship of MaxEnt to multinomial regression and log-linear models 68
4.4 Interpreting constraint weights 70
4.5 The math of weight-fitting 77
4.6 An introduction to priors 86
4.7 History and etymology of "MaxEnt" 92

Chapter 5: Research technique **94**

5.1 Assessment of analysis in MaxEnt 94
5.2 Graphing and visualization 94

5.3 Statistical assessment of MaxEnt grammars 97
5.4 On model-building 105
5.5 On obtaining probabilistic data 108

Chapter 6: The study of inventories 125

6.1 A MaxEnt theory of inventories 126
6.2 Vowel harmony as a phonotactic problem 128
6.3 Scaling up: the UCLA Phonotactic Learner 143
6.4 For further reading 143

Chapter 7: Modeling phonological learning 144

7.1 Some background: learning and learnability in MaxEnt 146
7.2 Non-veridical learning and its interest for UG study 148
7.3 Some biases 149
7.4 Modeling example: Hungarian vowel harmony 152
7.5 Modeling learning biases in MaxEnt: the use of priors 163
7.6 Bias modeling: some other studies 168
7.7 Bias modeling: the long term 169
7.8 Bias modeling: some other possibilities 170
7.9 Further topics in MaxEnt learning 171

Chapter 8: A survey of research areas to which MaxEnt grammars are applicable

8.1 Some research areas related to phonology 183
8.2 Research areas related to phonology 185
8.3 Metrics 189
8.4 Phonetics 189
8.5 Morphology 190
8.6 Syntax 190
8.7 Work blending syntax and phonology 194
8.8 For further reading 194

Chapter 9: MaxEnt as a theory of language 195

9.1 MaxEnt as probabilistic grammar 197
9.2 Constraints cannot be turned off 202
9.3 Intersecting constraint families and Magri's "four from three" principle 205
9.4 Do conjoined constraints subvert the predictions of MaxEnt? 209
9.5 If not MaxEnt, then what? 211

Chapter 1: Purpose and orientation

1.1 Rationale for this book

MaxEnt grammars (Smolensky 1986; Goldwater and Johnson 2003b) are a formal apparatus for constraint-based linguistics. MaxEnt as a grammatical model is an outgrowth of Optimality Theory (Prince and Smolensky, 1993/2004), but works with probabilities, making it suitable for dealing with gradient phenomena. These include free variation (multiple outputs from a single input), gradient well-formedness (e.g., in syntactic judgments or phonotactics), and the matchup of native speaker judgments to statistical patterns found in their language. MaxEnt grammars can also be used in providing concrete interpretations for linguistic experiments. Increasingly, linguists having been using MaxEnt for all of these purposes.

We will start by explaining why we have written this book. On various occasions, we have found ourselves conversing with linguists who have gathered interesting data that might have important implications for theory, yet are not in a position to construct a rigorous analysis of their data because they are quantitative in character. Our intent is to help such linguists overcome the barriers to using MaxEnt grammars for this purpose. MaxEnt offers both quantitatively precise analysis as well as statistical testing, which permits each constraint in the analysis to be checked for whether it is making a meaningful contribution.

We are phonologists and have primarily used phonological examples here. But we think linguists working in other branches of the field might also find the approach here helpful to their research. In Chapter 8: we survey examples from other areas of linguistics: phonetics, metrics, morphology and syntax, in which researchers have pursued a MaxEnt approach or something like it.

1.2 MaxEnt as tool vs. theory

We here consider MaxEnt from two perspectives. The first is the essentially practical one of our needing analytical tools to help us come to grips with quantitative linguistic data. We believe that MaxEnt can be employed by linguists to obtain closer, deeper understanding of their data: not only does MaxEnt derive precise, checkable quantitative predictions, but these predictions can be based on existing constraints that have the backing of typological and theoretical work. Our confidence in MaxEnt as a tool is based on our experience in guiding linguists who possess interesting data to use MaxEnt to understand their data better.

However, many linguists who have worked in MaxEnt have a more ambitious claim, namely that MaxEnt is a (partial) theory of how language works. Like OT, rule-based phonology, and many other linguistic theories, MaxEnt-as-theory aims to model the internal grammar of a speaker of a language, using tools that are proposed to be universal (the constraint set). In this way, we can extract from MaxEnt testable predictions about the cognitive architecture of language and about how that architecture shapes both language-specific phenomena and typology. Many of these predictions are inherited from OT; but others are MaxEnt-specific, discussed here in Chapter 9.

This distinction between tool and theory in part guides our exposition, which begins as a kind of how-to, and only gradually (Chapter 9:, but prefigured earlier), moves on to the theory's predictions and their assessment.

1.3 Why is it called "MaxEnt"?

One stumbling block for this theory is its name, "Maximum Entropy," which in the context of linguistics may come across as alien or even pretentious. This is not a problem we are in a position to fix, but in brief: "Maximum Entropy" in the context of linguistics designates the mathematical apparatus that was attached to Optimality Theory by Goldwater and Johnson (2003) to render it probabilistic. In current linguistics usage, "Maximum Entropy" or "MaxEnt" are often used to refer to the whole package; i.e. the OT architecture coupled with the MaxEnt math. The intellectual connection to actual entropy (as in physics or information theory) is, in practice, rather remote and perhaps best ignored at this stage. For further discussion see §4.6.

1.4 Content

The chapters of this book are arranged in the order which we believe will be most useful to the beginning practitioner. Chapter 2 offers an introduction to MaxEnt in the form of a tutorial, starting with a review of classical Optimality Theory. We illustrate both theoretical issues and the practicalities of how to conduct a MaxEnt analysis in an Excel spreadsheet, using a series of toy languages based on a lenition pattern in Khalkha Mongolian. The toy languages start simple and eventually work up to a complex pattern fairly close to what is observed in real Khalkha.

Chapter 3 provides a second worked-out example, this time using data from Tagalog. This case study is an example of **lexical frequency matching**, in which probabilistic patterns observed across the words of a language are productively applied to novel words by speakers in experiments. This type of data is broadly represented in the literature, and is often analyzed with MaxEnt. We conclude the chapter with an overview of frequency matching in general. By the end of Chapter 3, readers should feel well-equipped to develop MaxEnt analyses on their own.

Chapter 4 is devoted to mathematical issues. We first cover the details of the MaxEnt math, aiming to provide an intuitive understanding of what the math is doing and why. We also show how MaxEnt expands on (i.e. subsumes) classical OT. We next turn to the relationship of MaxEnt to statistics, particularly multinomial logistic regression. We show that while the statistical models and MaxEnt OT share the same math, their purposes and appropriate procedures are quite different. The chapter then moves on to various issues that have both mathematical and theoretical relevance, including how to interpret constraint weights, the issue of zero weights, infinite weights, and indeterminate weights, and priors. The chapter also includes a discussion of how the Excel Solver and similar systems compute weights. Lastly, we discuss the etymology and history of the term "MaxEnt."

Chapter 5 focuses on research methods. There are three sections: the evaluation of MaxEnt analysis with graphical and statistical methods; the construction of models (grammars) through the informed addition (or subtraction) of constraints from a possible constraint set; and lastly a

survey of ways in which a researcher might obtain probabilistic data in the first place: field notes, corpora, and experiments.

Chapter 6 covers the MaxEnt analysis of phonotactics, using Turkish vowel sequencing as an example. Phonotactic analyses illustrate many of the methods covered in Chapters 2 and 3, but tableaux are set up differently, with one large candidate set including all logically possible structures of interest. A good model should allocate probability to the structures which actually occur in the language, and away from the structures that are banned. The discussion also addresses an MaxEnt approach to the classical Subset Problem in language learning.

Chapter 7 is the main chapter addressing the theoretical problem of language learning. The key idea introduced is **bias modeling**: using regularization terms to bias certain constraints to have high or low weights. The idea is to use regularization to express a hypothesis about the language faculty (UG), which could help explain differences between speakers' behavior/intuitions and the patterns available to them during learning. We use Hungarian vowel harmony as our example, since this is a system where speakers match their lexicon on some generalizations but not others. At the end of the chapter we address two further issues from the MaxEnt perspective: hidden structure and gradual learning.

In Chapter 8 we give a brief overview of further possible uses of MaxEnt, including suggested readings for readers who wish to learn more. Within phonology, we discuss loanword adaptation, lexical strata, and underlying representation inference. Beyond phonology, we mention work in metrics, phonetics, morphology, and syntax.

The concluding Chapter 9 discusses specific predictions that MaxEnt makes as a theory of grammar. We take up empirical evidence for using probabilistic grammars in the first place, then move on to discuss predictions that are specific to MaxEnt, comparing it to other potential models of probabilistic data: Partially Ordered Constraints, Stochastic OT, Noisy Harmonic Grammar, and analogical models. In the process we also cover some critiques of MaxEnt.

1.5 Our MaxEnt spreadsheets and their locations (Supplementary Materials)

We are enthusiasts for the use of spreadsheets for MaxEnt calculations, favoring them for their transparency and ease of use. We find them particularly useful for advising graduate students: an adviser can jump-start the student's analysis by providing a small, conceptual spreadsheet on which the student can build.

We have placed a variety of spreadsheets discussed in the text on the internet, from which they can be downloaded; moreover, since websites can be fragile things, we offer the following places from which the spreadsheets are currently available:

https://brucehayes.org/
https://www.clairemoorecantwell.org/
https://osf.io/ (Open Science Foundation)

In the text we refer to the supplementary materials with boldface file names. To locate a file, visit the portion of the supplementary materials for the chapter you are reading, then search for the file name.

1.6 Policy on mathematical presentation

MaxEnt involves a certain amount of math. It can be used by a linguist as a tool without understanding any of it, but is more satisfying and reliable to the extent that one does understand the math. We address a broad linguistic audience here and seek to be helpful to readers with a range of mathematical backgrounds. To this end we have specifically targeted our writing to an imaginary reader who studied math in high school, and when reminded of various theorems and formulas taught there, can more or less call them to mind. We try to be helpful by showing and justifying all the steps of the mathematical derivations. This is impossibly verbose by the standards of a technical publication, but far more explicit than is given in (for example) popular science magazines. We beg the patience of readers whose mathematical background exceeds that of our hypothetical targeted reader. The mathiest sections have been placed in what we call Excursus Boxes; these can be skimmed if necessary but we do recommend at least giving them a try.

1.7 Acknowledgments

Many linguists have provided, and continue to provide, helpful comments and input on the preparation of this work. In particular, we would like to thank Adam Albright, Arto Anttila, Joan Bresnan, Jessica Coon, Gerhard Jäger, Gaja Jarosz, Tim Hunter, Aaron Kaplan, René Kager, Shigeto Kawahara, Stefan Keine, Giorgio Magri, Laurel Perkins, Robert Staubs, Megha Sundara, as well as the UCLA “Cans of Worms” reading group. Our apologies to anyone we may have forgotten. We would also like to thank the many graduate students over the years who have helped us hone our teaching of MaxEnt at UCLA. Special thanks to Corrina Fuller, who generously shared her Khalkha Mongolian data with us. We also thank Helen Barton at Cambridge University Press for all her help, as well as the six anonymous CUP reviewers who provided valuable comments on a draft version.

Chapter 2: The analysis of variation in outputs

We begin with a review of classical **Optimality Theory** (OT; Prince and Smolensky 1993/2004), then develop the exposition of MaxEnt by stages to cover, in the end, a nontrivial analysis. For pedagogical purposes we work with a sequence of imaginary languages, to be called *K1* through *K5*. These languages incorporate successively more complex data patterns, for which we employ first classical OT, then OT with free rankings (Anttila 1997), and finally MaxEnt itself.

The K1-K5 languages form successively better approximations to a real language, namely Khalkha Mongolian. For real Khalkha, our reference source is Fuller (2024), which includes substantial corpus data and its own detailed MaxEnt analysis. Our K1-K5 are intended to build up gradually, ultimately approximating the full complexity seen in Fuller's research. In §2.7.2 we briefly mention how real Khalkha goes beyond our hypothetical languages.

2.1 Analyzing the K1 language in Classical OT

The key (indeed, only) phenomenon of our first hypothetical language, K1, falls under the category of **lenition**, whereby stops and other obstruent sounds take on weakened degrees of closure. Lenition is a common process across languages (Lavoie, 2001; Gurevich, 2011; Kirchner, 2001); some classical instances include the lenition of Spanish /bdg/ to [βðɣ] (Harris 1969), or of English /td/ to the tap [ɾ] (Hayes 2008). In our hypothetical K1 language, we will suppose an underlying phoneme /p/ that in non-phrase-initial position is normally lenited to the homorganic glide [w], as in (1a) below. The pattern involves two complications. First, lenition is blocked whenever the /p/ is preceded by a nasal, as in (1b). Typologically, it is common for nasals (which, articulatorily, are stops) to disfavor nonstop articulations immediately after them; thus Spanish, with lenition of /b/ to [β] in many contexts, does not lenite /b/ after [m]; e.g. *la bomba* /la bomba/ → [la βomba] 'the bomb/pump'. Second, the lenition process of K1 is confined to function words (like [poɮ] 'become', or [pai] 'be'); when a /p/ appears in a content word, it never alternates; see (1c).

(1) *The patterning of /p/ lenition in K1*

a. Default realization in non-phrase-initial position: [w]

/mʊntəg **p**oɮ[1]-ən/ → [mʊntəg **w**oɮən]
great become-NPST

'will be great'

[1] The underlying form for 'become' with /p/ is justified by the appearance of [p] in other contexts, such as in /xun pəɮ-əx-ig/ → [xun pəɮəxig] (person become-FUT-ACC , *becoming a person*)

b. Retention of [p] after nasals

/ʃit-t͡ʃʰəx-sə**n** **p**ai-sən/ → [ʃitt͡ʃʰəxsə**n** **p**aisən]
decide-PERF-PST be-PST

'had decided'

c. Retention of [p] in content words

/ɔrʊɮ-ə**x** **p**ɔɮəm-t͡ʃtʰɔi / → [ɔrʊɮə**x** **p**ɔɮəmt͡ʃtʰɔi]
involve-FUT opportunity-have

'will have the opportunity to involve'

The pattern just given is the sort to which classical Optimality Theory (OT) has often been applied, and it reveals the basic ideas behind OT in action. We give here only the briefest review of classical OT. Readers who want better background in this theory are advised to consult one of the three excellent textbooks: Kager (1999), McCarthy (2002a), and McCarthy (2008). Much of what MaxEnt has to say reflects its status as an outgrowth of OT, and indeed many of the controversial issues for MaxEnt are controversial for OT as well.

In OT, the problem of phonological derivation is approached as searching for a particular **candidate output** that best satisfies a ranked hierarchy of phonological constraints. This **optimal** candidate is taken to be the (unique) output of the grammar. The theory assumes a function GEN that provides a full set of candidates — in the limit, the infinite set of all possible phonological strings. The function EVAL takes in a phonological input, the candidates of GEN, along with the ranked constraint set, and outputs a winner. Candidate selection works as follows: if the highest ranked constraint C that treats candidates $Cand_1$ and $Cand_2$ differently is violated more times by $Cand_2$ than $Cand_1$, then $Cand_2$ cannot be the winner. Similar comparisons, applied iteratively, will normally return a unique winning candidate.

We illustrate with K1. We will make use of four constraints, as follows. First, in order to account for the lenition in the first place, we will use the constraint *p:

(2) *p

Assign a violation to every non-phrase-initial [p].

This can be taken as one member of a larger family that penalizes segments that impose high articulatory effort, such as stops; see for instance Kirchner (2001)'s constraint LAZY.

If *p were the only constraint in the grammar, then of course all underlying /p/ in non-phrase-initial positions would be converted, often wrongly, to some other sound. But in K1 the process is "held back" in two particular contexts, namely after /n/ (1b) and in content words (1c).

We analyze the resistance to alternation in content words using the well-known family of **positional faithfulness** constraints (Beckman, 1997); for constraints distinguishing content from function words see Casali (1997) and Alderete (2003). Faithfulness constraints penalize candidates that change something going from input to output; and the standard taxonomy of Faithfulness constraints (McCarthy and Prince, 1995) is a systematic enumeration of all the

possible (minimal) changes that could affect a phonological representation. Here, for simplicity, we assume that [p] and [w] differ solely in the feature [sonorant],[2] and our crucial Faithfulness constraint will be called IDENT(sonorant)$_{content}$.

(3) IDENT(sonorant)$_{content}$

Output correspondents of an input [αsonorant] segment in a content word (as opposed to a function word) are also [αsonorant].

We can now begin to see the rationale for constraint ranking: *p alone will cause /p/ to be lenited everywhere except initially, but if *p is outranked by IDENT(sonorant)$_{content}$, lenition will not occur in content words, but will be confined to function words.

A comparable analysis can cover the other case of lenition blockage — after /n/ — discussed above. We use the constraint *N[+cont], from Coetzee (2000), which penalizes postnasal continuants. It penalizes the bad candidate *[nw] for underlying /np/; this is because [n] falls in the class [+nasal] and [w] is [+continuant].

(4) *N[+cont]

Nasal sounds should not be followed by a [+continuant] segment.

It is plain that *N[+cont] must dominate *p; or, to use the normal notation, *N[+cont] ≫ *p.

We now assemble the ingredients for an analysis: three constraints, plus schematic representative forms that, if properly derived, lead us to expect that the relevant forms of K1 can be derived in general. We are specifically interested in inputs where (a) /p/ is non-initial, but occurs in a content word; (b) /p/ is non-initial, but follows /n/; (c) /p/ is non-initial and neither of these two conditions are met. We want a form with [w] to be the winning candidate in case (c), but not in (a) and (b).

Aside from these constraints, we will also include in our analysis the constraint IDENT(sonorant) (i.e. the general case, *not* limited to content words). This constraint is needed to ensure that phrase-initial /p/ does not also lenite. We will not model this explicitly here, but include the constraint for completeness, and because it will become important later when variation is introduced. In general, careful analysis in OT includes constraints violated by "unfaithful winners" (McCarthy, 2008). It should be clear that *p must dominate IDENT(sonorant); else lenition to [w] would never happen at all.

We can now provide candidate tables, often called **tableaux** (singular "tableau"). At the top of each tableau, we list the constraints in an order consistent with the necessary pairwise rankings. Tableau rows give candidates from GEN, and asterisks indicate violations. It is easy for the eye to scan across such a tableau and observe how, going from left to right, the losing

[2] In the feature set of Hayes's (2008) text, they also differ in quite a few additional features: [voice], [round], [continuant], [consonantal], [approximant], [high], [back], [dorsal], and [tense]. A full analysis would include IDENT constraints for these features, as well as constraints expressing the impossibility of other possible outcomes such as [β] or [ʍ] that could repair a non-initial /p/.

candidates duly drop out of the competition, leaving the winner (indicated with the traditional "pointy finger") in place at the end.[3]

(5) *Tableaux for K1*

a. Simple lenition in a function word

/mʊntəg **p**oɻ-ən/ great become-NPST	IDENT(son)$_{content}$	*N[+cont]	*p	IDENT(son)
mʊntəg **p**oɻən			*!	
☞ mʊntəg **w**oɻən				*

b. Blockage of lenition after nasals

/ʃit-t͡ʃʰəx-sə**n** **p**ai-sən/ decide-PERF-PST be-PST	IDENT(son)$_{content}$	*N[+cont]	*p	IDENT(son)
☞ ʃitt͡ʃʰəxsə**n** **p**aisən			*	
ʃitt͡ʃʰəxsə**n** **w**aisən		*!		*

c. Blockage of lenition in a content word

/ɔrʊɻ-əx **p**ɔɻəm-t͡ʃtʰɔi / involve-FUT opportunity-have	IDENT(son)$_{content}$	*N[+cont]	*p	IDENT(son)
☞ ɔrʊɻəx **p**ɔɻəmt͡ʃtʰɔi			*	
ɔrʊɻəx **w**ɔɻəmt͡ʃtʰɔi	*!			*

We note that the rankings that were actually needed were really just *p ≫ IDENT(sonorant) (so lenition will happen); *N[+cont] ≫ *p (so lenition won't happen after nasals), and IDENT(sonorant)$_{content}$ ≫ *p (so lenition won't happen in content words). The constraints *N[+cont] and IDENT(sonorant)$_{content}$ never conflict with each other (both militate *against* lenition) and so there is no basis for assigning them a ranking (the order we pick for appearance on the page is arbitrary). The inessential ranking is shown with the dotted line in the tableaux of (5).

Our analysis thus far has been carried out using a relatively trivial GEN, with only two candidates for each input. A more valid analysis of K1, or any language, would select the right candidate out of an infinite GEN, and would require many more constraints and rankings. For example, in (5a) the candidate [mʊntəg oɻən], in which the offending [p] has been deleted rather than just lenited to [w], could be ruled out by ranking a MAX constraint (McCarthy and Prince 1995) over IDENT(sonorant). The more trivial candidate [mʊntəg əpoɻən], with an epenthetic schwa that does not help with the violation of *p, would be ruled out with a DEP constraint, as would even more bizarre candidates, like [ʊmʊntəg poɻən] or [mʊntəg poɻən**t**]. In actual

[3] We follow standard practice in other aspects of our tableaux: an exclamation point marks the particular asterisk (sometimes not first in its cell) that kills off a candidate. Gray shading indicates cells which are irrelevant to evaluation, because the winning candidate has already been selected somewhere to the left.

analytical practice scholars tend to present a GEN that includes anything that seems reasonable and bears on the problem at hand, but they must be on guard that they haven't missed something. Combinatorics, and software, can be helpful here (Karttunen, 2006; Riggle, 2009; Tesar, 2004). We discuss some issues of GEN specific to MaxEnt in §6.2 below.

OT has been extremely influential in phonological theory, adopted by many who had previously made use of the rule-based framework of *SPE* (Chomsky and Halle 1968). Before we go on, we mention two reasons why this is so.

First, OT accomplishes what many feel to be a scientifically sensible goal, namely stripping down the analysis of phonological systems to their minimal ingredients. Here, these are the Markedness principles that militate against particular surface forms and the Faithfulness constraints that atomize the set of ways in which surface forms can differ from their underlying forms. In contrast, rule-based theory deploys, in essence, *bundles* of Markedness-cum-Faithfulness: a rule like A → B / C___D combines a tacit "phonological problem" (a.k.a. Markedness constraint) that can be expressed as *CAD, together with the licensing of the change from A to B to solve this problem. (This point was first made in Prince and Smolensky, 1993:4). That such generality is worth achieving becomes evident when we note that the same Markedness constraint gives rise to different Faithfulness-violating remedies, both across languages (Kiparsky 1973) and across contexts within the same language (Kisseberth 1970). In other word, giving a distinct status to Markedness constraints solved the "conspiracy problem" that had baffled theorists starting in the 1970s.

Beyond this, a Markedness constraint sometimes can have the effect of *blocking* a change in some languages but *causing* it in another. Thus *N[+cont] blocks lenition in K1 and Spanish; but actively causes fricatives to become stops in languages with "postnasal hardening" (Rosenthall 1989).

A further benefit from OT's separation of Markedness problems from Faithfulness remedies is that we can make language-specific phonological analysis more responsible to typology: when things go well in OT (as they often do), we find that a very detailed language-specific analysis can be reduced to a language-specific ranking of principles each of which has broad cross-linguistic support. In the present case, we find that three principles all (with suitable library research) would be found to have extensive precedents beyond K1 (or more precisely, the real-life Khalkha on which K1 is based.) Stops lenite non-initially in a great number of languages, nasals frequently impede lenition, and content words often resist alternation more than function words do. Thus our analysis of K1 can be argued to have been assembled from "wholesome," typologically-sensible ingredients. At the very least, OT forces the analyst to declare which constraints in the analysis have broad cross-linguistic support and which ones are parochial; as we will see in Chapter 7, this distinction can have detectible consequences.

2.2 K2 and free variation

In our next hypothetical language K2, pretty much everything is the same, except that the process of /p/ lenition is optional. While it is possible for a native speaker of K2 to say [mʊntəg **w**oɮən] for item (1a), it is also possible to say [mʊntəg **p**oɮən]; and similarly for all similar

items. Since both are grammatical in the language, but no meaning difference arises, the phenomenon is commonly referred to as **free variation**.

Free variation is perhaps more common in language than some linguists (trained on problem sets) may be aware. The field of **variationist linguistics**[4] (Labov, 1969, 1973; Wolfram, 1993; Guy, 1993; Walker, 2010) has long studied free variation processes and achieved careful documentation of them in a variety of languages. Our own impression is that even in work with individual language consultants, free variation will often manifest itself provided the researcher is alert to the possibility — it often pays to ask the consultant "is also possible to say it this way?". Typically, carefully analyzed free variation is not a sea of chaos, but is phonologically structured (Cedergren and Sankoff 1974; Sankoff 1988), in ways that we will suggest are well suited to being addressed with MaxEnt grammars.

For the simplest cases of free variation, there is an immediately obvious possible approach, which occurred to theorists (notably Kiparsky, 1993, 2006ab; Anttila, 1995, 1997) almost as soon as OT was invented: instead of imposing a full ranking of the constraints in the grammar, we let certain constraint pairs (and perhaps triplets, etc.) be ranked freely. In the case of K2, we would posit free ranking for the constraints *p and IDENT(sonorant), as shown in the pair of tableaux in (6) below. Note that both tableaux retain the former free ranking of IDENT(sonorant)$_{content}$ and *N[+cont]— this is "trivial" free ranking, in that it has no effect on the outcome. The essential free ranking is that of *p and IDENT(sonorant), since when these are ranked differently, different candidates win.

(6) *A tableau pair for a case in which two constraints are freely ranked*

/mʊntəg **p**oɻ-ən/ (great; become-NPST)	IDENT(son)$_{content}$	*N[+cont]	*p	IDENT(son)
mʊntəg **p**oɻən			*!	
☞ mʊntəg **w**oɻən				*

PLUS

/mʊntəg **p**oɻ-ən/ (great; become-NPST)	IDENT(son)$_{content}$	*N[+cont]	IDENT(son)	*p
☞ mʊntəg **p**oɻən				*
mʊntəg **w**oɻən			*!	

This use of multiple tableaux is clumsy, and normally we just write one, with a dotted vertical line separating the freely ranked constraints, just as it does in (6) for IDENT(sonorant)$_{content}$ and *N[+cont].

[4] Also called **sociolinguistics**, especially in earlier work, though that term is now used more broadly. See Walker (2010) ch.1 for discussion.

(7) *A single tableau with free ranking that predicts multiple outputs*

/mʊntəg **p**oɻ-ən/ great become-NPST	IDENT(son)$_{content}$	*N[+cont]	*p	IDENT(son)
☞ mʊntəg **p**oɻən			*	
☞ mʊntəg **w**oɻən				*

Here in (7), we have just one pair of freely ranked constraints affecting the outcome, but the theory permits in principle any number of pairs (or triplets, quadruplets, etc).

2.3 The interpretation of free ranking as probability

The grammar illustrated in (7) is a rough-and ready account of free variation: it simply tells us that two outcomes are possible, accepting both [mʊntəg **p**oɻən] and [mʊntəg **w**oɻən]. Such a grammar would not make any predictions about which outcome was more common, and under what circumstances. However, in the work of Kiparsky (1993, 2006b, 2006a) and Anttila (1997), grammars like the one illustrated in (7) are used to predict precise probability distributions over candidate outputs.

This body of work proposes and refines an important and still-advocated theory of variation in OT. It has various names (Kiparsky, 1993; Anttila, 1997; Riggle, 2010), but we will call it the **Theory of Partially Ordered Constraints**, abbreviated "**POC**", following Coetzee and Pater (2011). To explain how the theory works, we use the example of tableau (7). Going across the constraints in this tableau from left to right, we let each group of constraints delimited by solid lines be called a *constraint stratum*. Thus for (7) the constraint strata are as in (8).

(8) *Constraint strata for tableau* (7)

a. *First stratum*

IDENT(sonorant)$_{content}$, *N[+cont]

b. *Second stratum*

*p, IDENT(sonorant)

Within each stratum (no matter how many constraints are in it), ranking under "POC" is assumed to be free. Further, it is assumed that any constraint in a higher stratum must outrank all constraints of a lower stratum. In order to predict probabilities over outputs, we first need to count the rankings compatible with these strata. Here, each stratum with two constraints in it gives rise to two possible rankings of those constraints; and because we have two strata each with two freely ranked constraints in them, the total number of rankings is 2 × 2 = 4. To be specific, the four rankings are as given in (9):

(9) *Rankings compatible with* (8)

a. IDENT(sonorant)$_{content}$ ≫ *N[+cont] ≫ *p ≫ IDENT(sonorant)
b. IDENT(sonorant)$_{content}$ ≫ *N[+cont] ≫ IDENT(sonorant) ≫ *p
c. *N[+cont] ≫ IDENT(sonorant)$_{content}$ ≫ *p ≫ IDENT(sonorant)
d. *N[+cont] ≫ IDENT(sonorant)$_{content}$ ≫ IDENT(sonorant) ≫ *p

In the general case, a stratum with *n* constraints gives rise to *n!* (*n* factorial, or n × (n − 1) × (n − 2) ...) possible rankings. We calculate n_i! for each freely ranked stratum *i*, then multiply out all of these terms to find the ranking count.

Now that we can count rankings, we can say that the **probability** assigned to any candidate under POC is simply the number of rankings for which that candidate is the winner, divided by the total number of rankings. The actual computation of this probability when the constraint set is intricate may be difficult, but the underlying concept seems straightforward. In the case of (7), there are four possible rankings (2! times 2!), of which two derive [mʊntəg **p**oɮən] and two derive [mʊntəg **w**oɮən].[5] Hence the prediction is that the lenited form [mʊntəg **w**oɮən] will surface exactly half the time, a probability of 0.5. Note that no ranking produces lenition in content words, or after nasals, so surface forms like *[ʃittʃʰəxsə**n** **w**aisən] or *[ərʊɮəx **w**ɔɮəmt͡ʃtʰɔi] are predicted to have a probability of zero.

What we have just established is sufficiently important that it is worth describing it in more general terms: unlike classical OT, a grammar under POC is a **probabilistic grammar**. Instead of just selecting one or more candidates from GEN as grammatical outputs for the given input, it assigns a probability value (often zero) to each member of GEN. It is probabilistic grammars with which we will concern ourselves in this book, as we seek to find analyses for K2 and other more complex cases.

POC is, arguably, the minimal extension of classical OT that can be used to create probabilistic grammars. POC grammars have been adopted for the analysis of a variety of systems with free variation (e.g. Pater 2000; Zuraw 2002). For the reasons we think POC is inadequate as a theory of gradient grammar, see §9.5.4 below; here, we move on directly to MaxEnt.

2.4 K3: modeling arbitrary probabilities

When researchers turn to quantitative examination of large datasets (texts, corpora, dictionaries, or datasets collected from speakers), it is common to find that variants are not equally likely as they were in K2; for discussion, see §9.1.

[5] Recall that the two constraints of the first stratum, IDENT(son)$_{content}$ and N[+cont], do not conflict, so that their ranking does not matter.

To explore this possibility, we introduce our third hypothetical language, K3, which will involve us with quantitative data. To give a sense of realism, we look ahead to the discussion (§2.7.2) of real Khalkha, as studied by Fuller (2024). Fuller obtained her data by carefully transcribing the phonetic realization of every instance of phonemic /p/ in TEDx talks (https://tedxulaanbaatar.com/), totaling about 53 minutes of speech, given by four native speakers of Khalkha in Ulaanbaatar, the capital city of Mongolia. From these data, Fuller collected 1334 instances of /p/, sufficient to shed light on the quantitative patterning of /p/ lenition in relatively formal Khalkha speech. Importantly, each individual talker in her data exhibited similar variation. Our synthetic data for K3 follow the same trends that Fuller observed, but have been constructed to be a maximally clear pedagogical example.

In K3, when a function word is preceded by a non-nasal sound, as in /mʊntəg **p**oɮ-ən/ above, our constructed data include 300 instances of [p] and 110 instances of [w]. Examples are given in (1). Crucially, surface [p], and surface [w] are not equally likely (as the tableau in (7) above, with partially-ranked constraints, would predict); in fact, the probability of weakened [w] is 110/(110+300) = 0.27. The remaining data are similar to K1 and K2, i.e. lenition never occurs in content words or after nasals.

(10) *Synthetic data for K3*

UR type	*Example*	*SR*	*count*
... [−nasal] [$_{\text{function}}$ p...	/mʊntəg **p**oɮ-ən/ great become-NPST	p	300 (73%)
		w	110 (27%)
... [+nasal] [p...	/ʃit-t͡ʃʰəx-sə**n** pai-sən/ decide-PERF-PST be-PST	p	212 (100%)
		w	0
... [$_{\text{content}}$ p...	/ɔrʊɮ-əx **p**ɔɮəm-t͡ʃʰɔi / involve-FUT opportunity-have	p	720 (100%)
		w	0

In order to model the data of K3, we seek a probabilistic grammar which can predict the exact lenition rates that we see in the data: 0% in content words and after nasals, and 27% in function words. This will consist of a MaxEnt grammar, so we first need to take a fairly substantial detour to explain what a MaxEnt grammar is.

2.5 MaxEnt: an introduction

Maximum Entropy Grammar is a variant of Optimality Theory, designed for the purpose of formulating probabilistic grammars. The key difference is that instead of using constraint ranking, MaxEnt assigns a numerical value to each constraint, its **weight**, which intuitively reflects its strength. We reemphasize here that in most respects MaxEnt is very much like OT (§4.2): it evaluates candidates in parallel, uses a potentially infinite GEN, and atomizes phonological analyses into simple, independent constraints; thus solving the conspiracy problem and linking language-specific analyses to phonological typology.

2.5.1 Preview: Classical Harmonic Grammar

To present MaxEnt clearly, we will begin with a simpler theory which will serve as a kind of way-station: **Classical Harmonic Grammar**, a theory originating with Legendre et al. (1990); for comparison with classical OT, see Pater (2009). In Classical Harmonic Grammar, the constraint weights and violations are used to calculate a **Harmony** ($\mathcal{H}$) score for each candidate, which (despite the name) acts as a kind of penalty. The output of the grammar in this theory is the least-penalized member of GEN for a given input. Note even though Classical Harmonic Grammar uses numbers, it is *not* a probabilistic theory: other than in rare cases of tied harmony, it produces one single winner for each candidate competition, just as in Optimality Theory.

A Harmony score is calculated as follows:

(11) *Calculating the Harmony of a candidate*

a. For each constraint, multiply the number of violations the candidate incurs by the constraint weight.
b. Sum the result obtained in step (a) across all constraints.

The same coputation is expressed as a formula in (12):

(12) *Formula: Harmony for candidate* x

$$\mathcal{H}(x) = \sum_{i \in Constraints} \mathrm{w}_i \times \mathrm{v}_i\,(x)$$

where $\mathrm{v}_i(x)$ is the number of violations incurred by candidate x for constraint i.

We illustrate with an example. Since Classical Harmonic Grammar allows only one single winner per input, we return here to the non-varying data of K1 ((1) above). The crucial tableau is (13), which is a Harmonic Grammar version of (5). For this tableau, we have hand-selected the weights for each constraint.

We begin with an example of Harmony calculation. Candidate [mʊntəg **p**oɮən] violates *p once. The weight of *p is 2, hence the Harmony contribution from this constraint for this candidate is 2. Since [mʊntəg **p**oɮən] *only* violates *p, its total Harmony score is therefore 2. Candidate [ʃittʃʰəxsə**n** **w**aisən] incurs one violation of *N[+cont], which has a weight of 3, and one violation of IDENT(sonorant), with a weight of 1. So its Harmony is 1 × 3 + 1 × 1 = 4.

Once we have calculated Harmony, it is straightforward to determine the unique winner for each input, namely the candidate with the lowest Harmony penalty.

(13) *Classical Harmonic Grammar tableau for K1*

	IDENT(son)$_{cont.}$	*N[+cont]	*p	IDENT(son)	
weights:	3	3	2	1	Harmony
a. /mʊntəg **p**oɮ-ən/ (great, become-NPST)					
mʊntəg **p**oɮən			1		2
☞ mʊntəg **w**oɮən				1	1
b. /ɔrʊɮ-əx **p**ɔɮəm-t͡ʃtʰɔi / (involve-FUT, opportunity-have)					
☞ ɔrʊɮəx **p**ɔɮəmt͡ʃtʰɔi			1		2
ɔrʊɮəx **w**ɔɮəmt͡ʃtʰɔi	1			1	4
c. /ʃit-t͡ʃʰəx-sə**n** pai-sən/ (decide-PERF-PST, be-PST)					
☞ ʃitt͡ʃʰəxsə**n** **p**aisən			1		2
ʃitt͡ʃʰəxsə**n** **w**aisən		1		1	4

In (13), each constraint is only violated once. To illustrate the multiplication of violation counts by weights, we consider another example from Fuller (2024), which contains multiple p's.

(14) *Multiple lenition sites in a single input*

Odoo	*khümüüs*	*oroi*	***untdag***	***polson***	***pai-na***
			/ʊntʰ-təg	pɔɮ-sən	pai-n/
now	people	late	sleep-HAB	become-PST	be-NPST

'Nowadays, people sleep late.'

The Harmony of neither, just one, or both /p/'s leniting is shown in (15):

(15) *Calculating Harmony in Classical Harmonic Grammar with multiple constraint violations*

	IDENT(son)$_{cont.}$	*N[+cont]	*p	IDENT(son)	
weights:	3	3	2	1	$\mathcal{H}$
/ʊntʰ-təg **p**ɔɮ-sən **p**ai-n/					
ʊntʰtəg **p**ɔɮsən **p**ain			2		4
☞ ʊntʰtəg **w**ɔɮsən **p**ain			1	1	3
ʊntʰtəg **p**ɔɮsən **w**ain		1	1	1	6
ʊntʰtəg **w**ɔɮsən **w**ain		1		2	5

As can be seen, for candidate [ʊntʰtəg **p**ɔɮsən **p**ain], the constraint *p contributes 4 units of Harmony, the product of two violations and its weight of 2. The winner here is actually [ʊntʰtəg **w**ɔɮsən **p**ain], with lenition on the first, but not the second, /p/. This one violates *p only once,

and IDENT(sonorant), which has a lower weight, once. This makes its Harmony just 3, lower than the 4 of the fully faithful candidate. Either candidate where the second p is lenited (the last two) also violates N[+cont], with its higher weight of 3.

2.5.2 Shifting to MaxEnt

The discussion of Classical Harmonic Grammar just given provided the mechanisms for converting weights and violation patterns into Harmony scores. For MaxEnt itself, there is one more step, namely to convert Harmony scores into probabilities. MaxEnt is a probabilistic version of Classical Harmonic Grammar, just as POC is a probabilistic version of OT. The math we will cover here will have two essential properties. First, it will assign lower probability to candidates that have higher Harmony scores. Second, the probability assigned to the full set of candidates for a given input will sum to one; this is what it means to be a probability distribution.

In (16), we give the MaxEnt transformation from Harmony to probability in the form of a recipe.

(16) *Calculating probability from Harmony values in MaxEnt*

a. Calculate the **Harmony** ($\mathcal{H}$) of each candidate, as in (12).

b. For each Harmony value, first make it negative, and then exponentiate it using Euler's number (*e*).[6] We'll call this value **eH**:

$$\mathrm{eH} = e^{-\mathcal{H}}$$

c. For each input, sum up the eH over all the candidates for that input. Call this **Z**.

$$Z = \sum_{i \in candidates} \mathrm{eH}_i$$

d. For each candidate, divide its eH by Z. This is its predicted **probability**.

$$p(i) = \frac{\mathrm{eH}_i}{Z}$$

To illustrate these mechanisms, we return to K3, the language in which [mʊntəg **p**oɮən] occurs 73% of the time and [mʊntəg **w**oɮən] 27%. We seek to formulate a MaxEnt grammar that matches this percentage, and also derives the invariant outcome for the remaining two inputs.

Formally, a MaxEnt grammar consists of a set of constraints, each paired with a weight. Our proposed grammar is given in (17).

[6] Euler's number *e*, useful in mathematics for many reasons, is about 2.718. MaxEnt would work perfectly well with some other base, such as 10, but we are following standard practice here.

(17) *A MaxEnt grammar for K3*

Constraint	*Weight*
*p	3
IDENT(son)	4
IDENT(son)$_{content}$	20
*N[+cont]	20

The weights in (17) were chosen by hand. While it is possible to find a good fit to the data by choosing weights by hand for a very small dataset like this one, realistic analyses generally require the weights to be fit by machine, a topic taken up in §2.6.

For now we will simply accept the weights of (17) and proceed with the computation. All the needed values appear in tableau (18) below, which may also be consulted in spreadsheet form in the supplementary materials as **K3.xlsx**. The weights of (17) are shown in (18) underneath their respective constraint names; and the constraints assign violations in the same way as in previous tableaux.

To illustrate the calculations, we will derive step by step the probability of candidate [mʊntəg **p**oɮən], which competes with [mʊntəg **w**oɮən] as the output for /mʊntəg **p**oɮ-ən/. First, its Harmony is calculated in the same way as in Classical Harmonic Grammar; it comes out as 3 × 1 = 3. To convert this Harmony value into a probability, we follow the procedures in (16). eH, following (16b), comes out to e^{-3} = 0.050. Next, (16c) tells us to calculate Z by summing eH values for both candidates. Here, we calculate eH for the competitor candidate [mʊntəg **w**oɮən], which comes out to 0.018, then add the two eH values to obtain 0.067. Lastly, we calculate the probability of [mʊntəg **p**oɮən] according to (16d), by dividing its eH by Z. This is 0.050/0.068 = 0.73. The last three columns of tableau (18) illustrate the computations for all candidates.

(18) *Tableau: MaxEnt analysis of K3*

/mʊntəg **p**oɻ-ən/ (great; become-NPST)	Freq.	prob.	IDENT(son)$_{content}$	*N[+cont]	*p	IDENT (son)	$\mathcal{H}$	eH	Z	p
		weights:	20	20	3	4				
mʊntəg **p**oɻən	300	0.73			1		3	0.050	0.068	0.73
mʊntəg **w**oɻən	110	0.27				1	4	0.018	0.068	0.27
/ɔrʊɻ-əx **p**əɻəm-t͡ʃʰɔi / (involve-FUT; opportunity-have)										
ɔrʊɻə**x** **p**əɻəmt͡ʃʰɔi	720	1			1		3	0.049	0.050	≈1
ɔrʊɻə**x** **w**əɻəmt͡ʃʰɔi	0	0	1			1	24	3.77×10^{-11}	0.050	≈0
/ʃit-t͡ʃʰəx-sə**n** pai-sən/ (decide-PERF-PST; be-PST)										
ʃitt͡ʃʰəxsə**n** **p**aisən	212	1			1		3	0.049	0.050	≈1
ʃitt͡ʃʰəxsə**n** **w**aisən	0	0		1		1	24	3.77×10^{-11}	0.050	≈0

As you can see, we have chosen our weights such that the MaxEnt-predicted probabilities closely match the frequencies of our toy data set, given as raw numbers in the second column of (18) and as proportions in the third. The generation of probabilities like 0.73 is beyond the capacity of Classical OT or Classical Harmonic Grammar, which are nonstochastic theories. Using POC (§2.3) would also be insufficient, since only a probability distribution of 50/50 can be generated.[7] MaxEnt, in contrast, can generate any probability distribution in such cases with a suitable choice of weights.

At this point we have illustrated the core of MaxEnt, the math that converts Harmony ($\mathcal{H}$) to probability. To wrap things up, we reexpress the MaxEnt recipe of (16) as a single mathematical formula, given in both terse and verbose form; the latter spells out the Harmony calculation.

[7] For further discussion see §9.5.4.

(19) *Maxent formula*

a. Terse

$$P(x) = \frac{e^{-\mathcal{H}(x)}}{Z} \quad \text{where } Z = \sum_i e^{-\mathcal{H}(\text{cand}_i)}$$

Key: x is a candidate for some input
$P(x)$ is the probability the grammar assigns to x
e^t is e taken to the tth power
$\mathcal{H}(x)$ is the harmony of candidate x
$cand_i$ is the ith candidate for the given input
Σ_i denotes the sum over all candidates

b. Verbose

$$p(x) = \frac{e^{-\sum_{i \in Con} w_i \times v_i(x)}}{\sum_{y \in candidates} e^{-\sum_{i \in Con} w_i \times v_i(y)}}$$

w_i is weight of the ith constraint
$v_i(y)$ is the number of times candidate y violates the ith constraint
Σ_i denotes the sum over all constraints

Before going on, we offer a precaution to the reader: in the literature one will find variants of (19) that are essentially the same calculation. In particular, if we put negative signs in front of every violation, as in Coetzee and Pater (2011), then Harmony will come out negative, and we can omit the negation step in (16b). The same will hold true if we negate the constraint weights. Our own practice (positive weights to express constraint strength, violations counted in the ordinary way) derives from the original paper of Goldwater and Johnson (2003). In addition, the very word "Harmony" sometimes is used to designate something else; for instance Magri (2025) uses "Harmony" to designate what we call eH. These variants will generally be intelligible in context.

2.6 K4: Perturbers as "hidden" phonology in variation

We now move to our next hypothetical language, K4. This toy language is intended to represent a very common situation in patterns of variation. Not only is the probability of the two possible outcomes unequal, but additionally it is *affected by context*; that is, there is at least one independent factor — we will call it a **perturber** — that changes the output probabilities in a systematic way (Labov 1969, Bayley 2002). In a MaxEnt grammar, this additional factor will usually be treated using what will call a **perturber constraint**. Candidates that violate the perturber constraint will have a probability distribution different from what is seen among non-perturber-violating candidates.

Our K4 language is set up to illustrate the effect of a perturber. As in K3, lenition does not occur in content words or after nasals. However, in other environments the lenition rate is

affected probabilistically by the preceding context: specifically, lenition is more likely after a vowel than after a consonant. For example, in an input like /mʊntə**g p**oɻ-ən/, with preceding consonant, lenition occurs about 17% of the time. But in /tʊtʰ**ʊ p**ai-ga/, with a preceding vowel, lenition occurs considerably more often, 83% of the time. Hence a preceding vowel is a perturber that increases the probability of lenition. The pattern is summarized in (20).

(20) *Synthetic data for K4*

UR type	Example	SR	*count*
... C [function p ...	/mʊntəg **p**oɻ-ən/ great become-NPST	p	*290 (83%)*
		w	*60 (17%)*
... V [function p ...	/tʊtʰʊ **p**ai-ga/ missing be.PROGPRES	p	*10 (17%)*
		w	*50 (83%)*
... [content p ...	/ɔrʊɻ-əx **p**ɔɻəm-t͡ʃʰɔi / involve-FUT opportunity-have	p	*720 (100%)*
		w	*0*
...[+nasal] [p ...	/ʃit-t͡ʃʰəx-sə**n** pai-sən/ decide-PERF-PST be-PST	p	*212 (100%)*
		w	*0*

Our K4 language illustrate a very common situation when researchers examine probabilistic phonological patterns (whether this be lexica, corpora, or experiments): While at first it may appear that all varying forms pattern together (as was true in K3), further inspection reveals non-uniformity: after a vowel, we have predominant lenition (83%), whereas after a consonant lenition is far less usual (17%). If we did not examine the specific rates of lenition in these two contexts, but rather pooled them together, we would have failed to notice the effects of the perturber environment.

To formalize the perturber effect in K4, we adopt a commonplace lenition constraint as our perturber constraint, defined in (21):

(21) POSTVOCALICLENITE

Assign a violation for every sequence of the form [+syllabic][−sonorant].

The idea here is that, as in in many languages, K4 disprefers postvocalic stops, and replaces them with a more sonorous consonant, in this case [w].

For the moment, we stick with the constraint weights we used for K3, but add in POSTVOCALICLENITE, with a weight of 2, chosen by hand. The tableaux in (22) illustrate the predictions of this new grammar. The new input /tʊtʰʊ **p**ai-ga/ is now included, for which the POSTVOCALICLENITE constraint affects the predicted probability distribution.

(22) *Tableaux for K4 (hand-fit weights)*

a. Simple leniting environment (no perturber)

/mʊntəg **p**oɮ-ən/
great become-NPST

	Freq.	prob.	IDENT(son)content 20	*N[+cont] 20	*p 3	IDENT (son) 4	POSTVOCLENITE 2	$\mathcal{H}$	eH	Z	p
mʊntəg **p**oɮən	290	0.829			1			3	0.050	0.068	0.73
mʊntəg **w**oɮən	60	0.171				1		4	0.018	0.068	0.27

b. Leniting environment with postvocalic perturber

/tʊt^{h}ʊ **p**ai-ga/
missing be.PROGPRES

tʊt^{h}ʊ **p**ai-ga	10	0.167			1		1[8]	5	0.007	0.025	0.27
tʊt^{h}ʊ **w**ai-ga	50	0.833				1		4	0.018	0.025	0.73

c. Blockage in stems

/ɔrʊɮ-əx **p**ɔɮəm-t͡ʃt^{h}ɔi /
involve-FUT opportunity-have

ɔrʊɮə**x** **p**ɔɮəmt͡ʃt^{h}ɔi	720	1			1			3	0.050	0.050	≈1
ɔrʊɮə**x** **w**ɔɮəmt͡ʃt^{h}ɔi	0	0	1			1		24	3.77* 10^{-11}	0.050	≈0

d. Blockage after nasal

/ʃit-t͡ʃ^{h}əx-sə**n** pai-sən/
decide-PERF-PST be-PST

ʃitt͡ʃ^{h}əxsə**n** **p**aisən	212	1			1			3	0.050	0.050	≈1
ʃitt͡ʃ^{h}əxsə**n** **w**aisən	0	0		1		1		24	3.77* 10^{-11}	0.050	≈0

Our predicted probabilities are reasonably close to the observed ones. (We will see in the next section that fitting the weights by algorithm obtains a closer fit.) The key point for now is that we have successfully incorporated the effect of POSTVOCALICLENITE as a perturber constraint: it lowers the output probability of candidates that violate it (here 0.27 for [tʊt^{h}ʊ **p**ai-ga] vs. 0.73 for [mʊntəg **p**oɮən]), but does not rule out violating candidates entirely.

2.6.1 More on perturber constraints

It is common for constraints that act as a perturber in one language to appear as a fully determinative factor in another language. For example, Spanish is said to be a language in which post-vocalic lenition (of voiced stops) is entirely obligatory: *navidad* /nabidad/ 'Christmas' *must* be [naβiðað], and cannot realize any of the obstruents as a stop. Although our use of postvocalic

[8] Note that since POSTVOCALICLENITE is assessed for all segment pairs, there should actually be actually many violations present throughout the tableaux. We record only the violations that distinguish candidates; the outcomes remain the same.

lenition as a perturber in Khalkha is invented, we cover in §9.1.3 several real-language pairs of phenomena where a factor that fully determines the outcome in an obligatory process in one language or languages acts as a perturber in another language. To give just one example here, in Iraqi Arabic a ban on triple consonant clusters (*CCC) is categorical: when such sequences arise across morpheme boundaries, they are obligatorily repaired (Broselow, 1980). But in English, the same constraint acts as a perturber; it merely increases the frequency with which CCC clusters are repaired in surface forms. For example, the final /t/ of *tanked* [tæŋkt] is deleted more often than the final /t/ of *tact* [tækt] (Labov, 1989). In the context of a constraint-based theory like MaxEnt, this pretheoretical observation has a clear theoretical interpretation, namely, that variable and categorical phenomena use the same constraint set. Obligatoriness vs. optionality is simply a consequence of the particular weights that have been assigned.

The pervasive appearance of perturbers in variation is perhaps the best argument for the use of MaxEnt or similar frameworks as an analytical tool: without creating a probabilistic analysis of the data, one is liable to miss important parts of the language system. In Classical Optimality Theory, a nonstochastic theory, there is no way to treat perturber phenomena — it was simply not designed for this purpose. MaxEnt gives us a way to integrate perturber phenomena into language-specific analysis and, indeed, into the study of typology.

We also suggest that MaxEnt might also be of assistance to the working linguist in trying to make sense of new data: it encourages the analyst to seek out gradient generalizations and test them through explicit formal analysis.

2.6.2 Fitting the weights

Establishing the weights is, of course, part of the MaxEnt analysis, analogous to finding a workable ranking in Optimality Theory. However, weighting is different from ranking in that it is almost impossible to do it accurately by hand. We have found it a useful classroom exercise in teaching beginning MaxEnt to ask our students to do hand-weighting; we find that it is possible but generally difficult, and usually leads to a less accurate grammar than we can obtain by using a machine-implemented algorithm. Following standard usage, we use the term **fitting** to denote the process of finding the best available weights for a given set of data and constraints.

MaxEnt weights can be fit by algorithm using a variety of different software types, which we cover briefly in §2.11. For the MaxEnt beginner, we highly recommend the use of Excel for implementing the MaxEnt math, and the Excel Solver add-on for fitting weights. We recommend this approach mainly for transparency. The cells of the spreadsheet display in full the math that calculates harmony and converts it to probability. Furthermore, the Excel format makes it easy to implement changes to the analysis and understand their consequences. More experienced MaxEnt practitioners may also find the Excel format useful, as it makes it easy to introduce novel elements to the structure of the theory. We will illustrate many examples of this throughout the later chapters of the book. Lastly, Excel sheets can be placed into online Supplementary Materials of a published paper, so that the calculations become freely inspectable, and modifiable, by anyone with access to Excel.

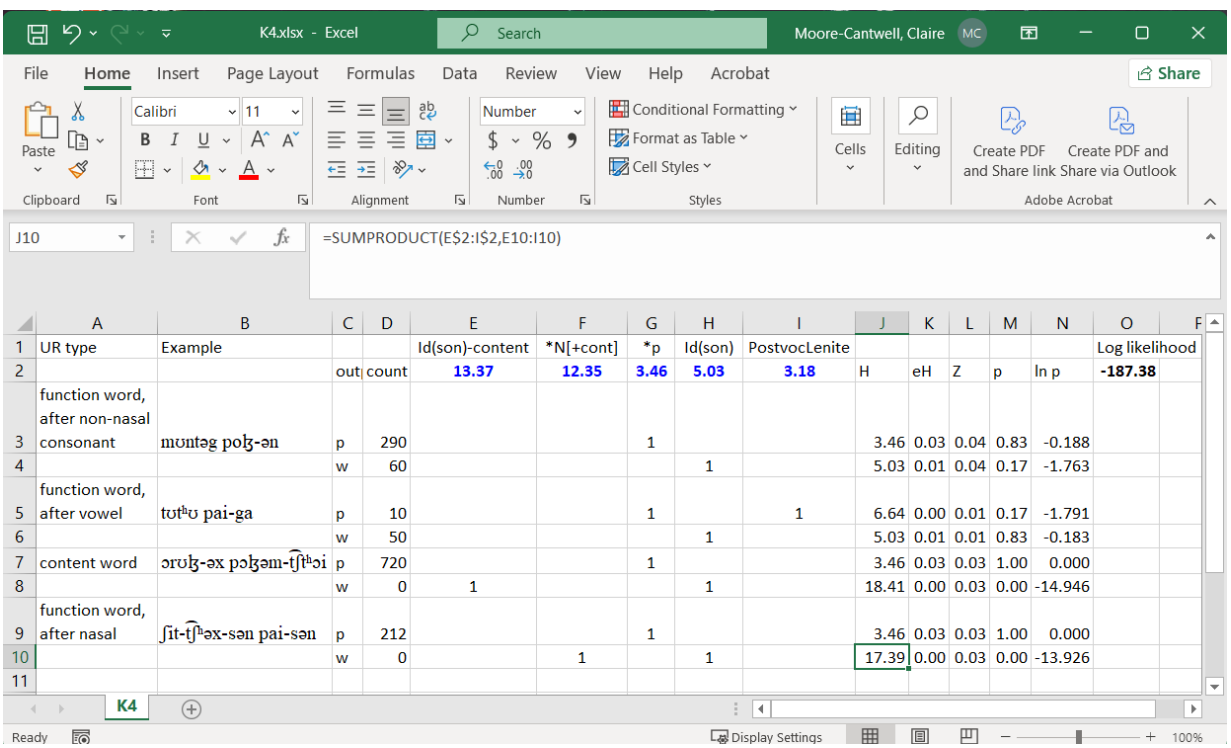
What follows is a "how-to" for implementing a MaxEnt analysis in Excel. We first show how to implement the MaxEnt formula (19) in the format of an Excel spreadsheet (§2.6.3). With this in place, we show how to calculate the weights (§2.6.4).

2.6.3 Implementing the MaxEnt formula on a spreadsheet

The first step is to implement the MaxEnt formula, given above in (19), using the resources of Excel. For this purpose we return to our K4 example. The spreadsheets discussed here may be obtained in the Supplementary Materials for this book, available at the web locations listed in §1.5; the particular spreadsheet we will be working with can be found among the Chapter 2 files as **K4.xlsx**. This spreadsheet is given in print below, and you may find it helpful not to download it but rather to type it out yourself as a study aid.

Figure 1 is a screen clip of the actual Excel interface. It is opaque, as one sees the "underlying" values of the cells only when you hover the mouse over them. The content of cell J10 has been selected and thus shows up in the formula bar (large white window).

Figure 1: The Excel interface



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	UR type	Example			Id(son)-content	*N[+cont]	*p	Id(son)	PostvocLenite						Log likelihood
2			out	count	13.37	12.35	3.46	5.03	3.18	H	eH	Z	p	ln p	-187.38
3	function word, after non-nasal consonant	mʊntəɡ pokɟ-ən	p	290			1			3.46	0.03	0.04	0.83	-0.188	
4			w	60				1		5.03	0.01	0.04	0.17	-1.763	
5	function word, after vowel	tʊtʰʊ pai-ga	p	10			1		1	6.64	0.00	0.01	0.17	-1.791	
6			w	50				1		5.03	0.01	0.01	0.83	-0.183	
7	content word	ɔrʊkɟ-əx pəkɟəm-t͡ʃʰʰɔi	p	720			1			3.46	0.03	0.03	1.00	0.000	
8			w	0	1			1		18.41	0.00	0.03	0.00	-14.946	
9	function word, after nasal	ʃit-t͡ʃʰəx-sən pai-sən	p	212			1			3.46	0.03	0.03	1.00	0.000	
10			w	0		1		1		17.39	0.00	0.03	0.00	-13.926	
11															

The table in (23) shows the underlying structure of Figure 1, in particular the formulae that compute the displayed values.[9] We begin at column B, which contains the example inputs of each type (they appear shortened in (23) relative to the actual spreadsheet). Column C contains the candidate outputs, and column D contains the counts of each candidate output 'observed' in

[9] For specific Excel formulae, we find the "Help" menu within Excel itself can be useful. Many training resources for Excel exist online.

our toy data. Columns E-I contain constraint names in row 1 and weights in row 2. Constraints in columns E-G are not shown, but in the full version of the spreadsheet are as in tableau (22).

(23) *Calculating MaxEnt probabilities in Excel*

Excel column names

row #	B	C	D	...	H	I	J	K	L	M
1	Example			...	Id(son)	Postvoc Lenite	Harmony	eH	Z	p
2		Output	count		4.00	2.00				
3	g poʒ-	P	290				=SUMPRODUCT (E$2:I$2, E3:I3)	=EXP(-J3)	=SUM(K3:K4)	=K3/L3
4		W	60		1		=SUMPRODUCT (E$2:I$2, E4:I4)	=EXP(-J4)		=K4/L3
5	ʊ pai-	P	10			1	=SUMPRODUCT (E$2:I$2, E5:I5)	=EXP(-J5)	=SUM(K5:K6)	=K5/L5
6		W	50		1		=SUMPRODUCT (E$2:I$2, E6:I6)	=EXP(-J6)		=K6/L5
...	...	...	...	...	...	...	...	...	...	...

The next few columns implement the MaxEnt math, closely following the procedure of (16). Column J calculates the Harmony of each candidate, rendering in Excel the formula from (12), $\sum_{i \in Constraints} w_i \times v_i$(candidate). It uses the Excel function `SUMPRODUCT()`, which performs two steps at once: it multiplies together the individual values for constraint violation counts and weights (columns E-I), then sums the result across all constraints. The expression "`E$2:I$2, E3:I3`" in cell `J3` follows standard Excel conventions: `E$2:I$2` means "always the cells in row 2", hence always the row of weights; but "`E3:I3`", since it appears in row 3, means "whatever appears in cells E-I of this row". By using this notation, it is possible to write the function once, in the top cell (here, `J3`) then copy it down to fill the rest of the column. Excel will automatically update any contextual references, so that the constraint weights will be multiplied by whatever violation counts are present in a given row.

Column K computes the eH value of each candidate *x* (i.e. $e^{-\mathcal{H}}$, from (16b)) by first negating, then exponentiating, the result of Column J. As can be seen, the Excel notation for e^x is `EXP(x)`. As before, this value can be written once and copied down the column.

Column L calculates the normalizing factor Z for each input (16c), using the `SUM()` function to total the eH values for all (both) candidates.

Column M carries out the final step, from (16); i.e. calculating the probability assigned to each candidate by dividing the eH from column K by the Z value from the appropriate cell in column L.

2.6.4 Setting the weights by algorithm

Now that we have expressed the MaxEnt calculations in a spreadsheet, we can turn to our original problem of using software to find the most accurate constraint weights.

To begin, we first have to quantify what we mean by "accurate." We will employ a widely-used metric of accuracy: the **log likelihood** assigned by the grammar to the data (Jurafsky and Martin, 2026). Log likelihood is a measure of how likely our observed data are, given our current set of weights. It is defined as the log of the probability of the observed data if the model is assumed to be correct. A higher log likelihood value implies greater accuracy. The search for optimal constraints is now defined as finding the weights that maximize log likelihood. We delay detailed discussion of the rationale for log likelihood to §4.5 below; here we simply give the steps that are needed to calculate it in Excel.

The procedure for log likelihood calculation in Excel is illustrated in (24), which is extracted from the **K4.xlsx**; we show the additional columns which are used to calculate log likelihood. Column M, already seen, gives the predicted probability for each candidate. Column N converts this probability to log probability, using the log function (`LN()`) in Excel. Since many of the data types occur multiple times in the K4 data (see column D), we take account of each datum by multiplying these log probability values by the frequencies, using the convenient `SUMPRODUCT()` function, which we used earlier to calculate harmonies in column J. The result is given in cell `O2`.

(24) *Calculating log likelihood in Excel*

Excel column names

row #	B	C	D	...	H	I	...	M	N	O
1	Example			...	Id(son)	Postvoc Lenite	...	p	log p	Log likelihood
2		output	count		4.00	2.00				=SUMPRODUCT(D3:D10,N3:N10)
3	g poɣən	P	290					=K3/L3	=LN(M3)	
4		W	60		1			=K4/L3	=LN(M4)	
5	ʊ pai-	P	10			1		=K5/L5	=LN(M5)	
6		W	50		1			=K6/L5	=LN(M6)	
...	...	...	...	...	...	...	...	...		

Note that some candidates are never observed; i.e. their frequency is zero; because of the multiplication, these candidates do not contribute to log likelihood.

When we view **K4.xlsx** in the normal mode of Excel, we see the values computed by the formulae in (24):

(25) *Same table, with displayed values*

	Excel column names									
row #	B	C	D	...	H	I	...	M	N	O
1	Example			...	Id(son)	PostvocLenite	...	p	log p	Log likelihood
2		output	count		4.00	2.00				-198.44
3	g poɣən	p	290					0.73	-0.313	
4		w	60		1			0.27	-1.313	
5	ʊ pai-	p	10			1		0.27	-1.313	
6		w	50		1			0.73	-0.313	
...	...	...	...	...	...	...	...	...		

As you can see, the log likelihood, −198.44, is a negative number, since the log of a probability (something less than 1) is always negative. Our goal is to maximize this value, specifically to find a negative value that is as close as possible to zero.

Now that we have a cell that calculates log likelihood, we can use it as our accuracy metric, guiding the search for the best weights. This can be carried out in Excel using the "Solver" utility, which comes with the program but must be activated.[10] Unfortunately, although Solver programs exist for Google Sheets and Libre Office, at the time of this writing these versions do not contain the algorithm necessary for fitting a MaxEnt model. [xxx check this one more time before publication.]

To find the best weights, we recommend carefully following the following steps:

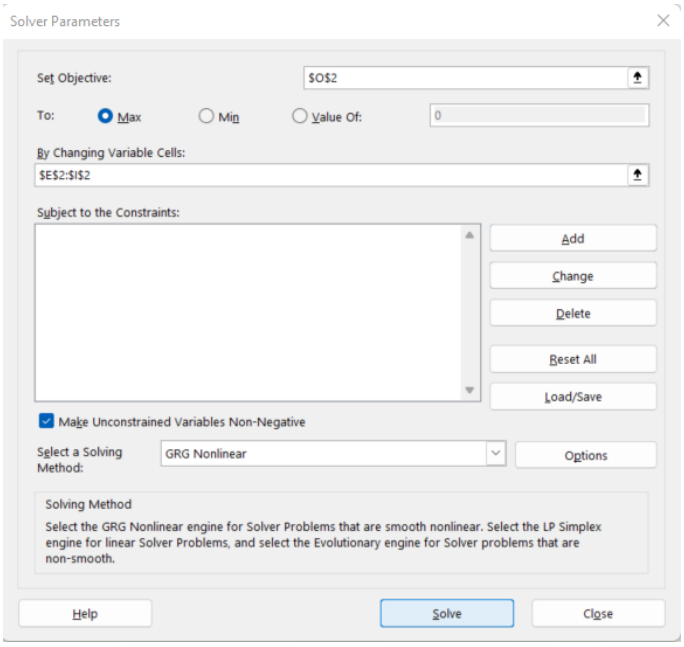
1) **Initialize:** Set all constraint weights to 0 (in **K4.xlsx**, this would be cells `E2:I2`).
2) **Select Log likelihood as objective:** Open the Solver interface, and select the cell where you have calculated log likelihood (here, cell `O2`) for the "Set Objective" field. Since we are trying to maximize log likelihood, you should select the "Max" button.
3) **Select weights:** Select the cells which contain your weights (cells `E2:I2`) in the "By changing variable cells" field.
4) **Keep weights non-negative:** In order to avoid negative weights, make sure that the checkbox next to "Make Unconstrained Variables Non-Negative" is checked.
5) **Select solving method:** Where it says "Select a solving method", choose "GRG Nonlinear" (usually the default).[11]

[10] In the version of Excel we are currently using the procedure is: File > Options > Add-ins. In the Manage drop-down menu, select "Excel Add-ins" and click Go. Then, check the box next to "Solver Add-in" and click OK. [xxx check before publication] On Mac it is under Tools [xxx fill out]

[11] GRG Nonlinear in general terms can be studied in Lasdon et al. (1974), though the specific implementation in the Solver is proprietary. GRG Nonlinear is just one of many algorithms that can be used to effectively compute MaxEnt weights; the choice of algorithm is in most cases a purely practical matter and will not affect the analytic results obtained; see §4.5.2.

The settings of all these conditions for the Solver are shown in the screen clip of Figure 2.

Figure 2: Launching the Solver on the K4 problem



With these conditions set, one can click on the Solve button, and for at least for schematic problems of the size described here, weights are found quite quickly. These weights are as follows:

(26) *Constraint weights for the K4 analysis*

IDENT(son)$_{content}$	13.37
*N[+cont]	12.35
*p	3.46
IDENT(son)	5.03
AGREE-round	3.18

These weights provide a close fit to the observed probabilities in K4, as (27) shows.

(27) *Observed vs. predicted probabilities for the K4 analysis*

Input (see Fig. 1, column A)	*candidate*	*observed probability*	*predicted probability*
function word, after non-nasal consonant	p	0.829	0.828
	w	0.171	0.172
function word, after vowel	p	0.167	0.167
	w	0.833	0.833
content word	p	1.000	1.000
	w	0.000	0.000
function word, after nasal	p	1.000[12]	1.000
	w	0.000	0.000

Thus we now have a complete MaxEnt analysis of K4, which fits the data quite well. All steps in the creation of the analysis have been described explicitly, with the one exception of the GRG Nonlinear algorithm, internal to the Solver, that locates the sets of weights under which log likelihood has been optimized; this alone has been left as a black box.

2.7 K5: Multiple perturbers

Having introduced the basics of MaxEnt grammar via the toy languages K3 and K4, we now turn to a final toy language, which is closely based on the real Khalkha data analyzed by Fuller (2024). This more complex example will illustrate a more intricate type of data pattern that shows up widely in MaxEnt analysis.

First, a trivial change: we will treat the post-nasal environment not as an absolute blocker of lenition, but merely a downward perturber; i.e., forms like /ʃittʃʰəxsə**n p**aisən/, from K4, do occasionally undergo lenition.

More fundamentally, in K5 we adopt a crucial observation made by Fuller (2024), who observed different rates of lenition in different function words. In *pi*, the first person pronoun, she observed almost no lenition (just 7 lenited forms out of 191). On the other hand for the particle *pol*, she observed 112 cases of lenition out of 163. She observed different rates in two other function words as well, one of which is homophonous with the particle *pol*. In this case we adopt for K5 some values actually taken from Fuller.

[12] Note that these predicted probabilities are not exactly 1 and 0, but rather very close to 1 and zero; close enough that Excel displays only 1 or 0.

(28) *Data from Fuller (2024), figure 10*

Function word:		*Instances with p*	*Lenited instances*	*Lenition rate*
pol	particle used for interrogatives, conditionals, and focus	51	112	68.7%
pol-	copula, "to become"	55	57	50.9%
pai	copula, "to be"	142	99	41.1%
pi	1st person pronoun	184	7	3.7%

Fuller found that these preferences of each function word coexist alongside the grammatical effects of blockers (the nasals in K4) and of the lenition-preferring environment (post-vowel in K4). Each function word, though it has its own baseline probability of leniting, lenites more in the lenition-preferring environment, and less in the blocking environment.

In the present case, morpheme identity can be treated as another type of perturber, in this case lexical rather than phonological. If, hypothetically, we think of the lenition rate for *pol* as a sort of default, then using the other morphemes *pol-, pai*, or *pi* in effect perturbs the lenition rate downward.

To give the full data set, we cross-classify the effects of phonological environment already seen with the effect of morpheme identity, as in (29). The counts are based on Fuller's actual corpus data, but simplified somewhat.

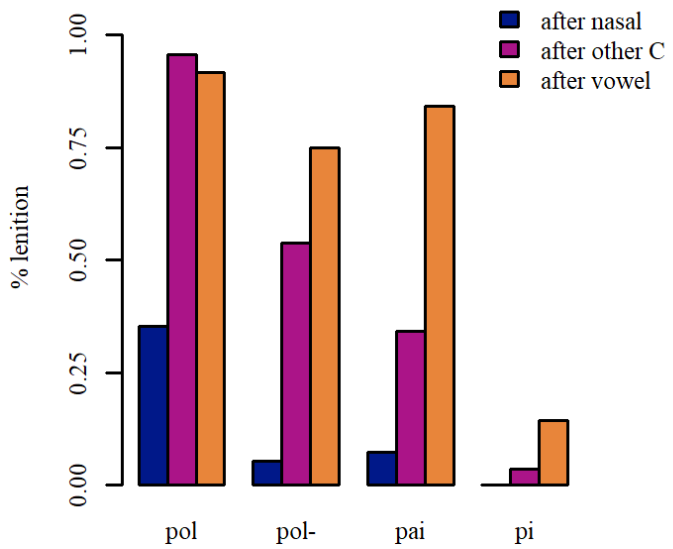
(29) *The K5 data pattern (**K5.xlsx**)*

Function word	*UR type*	*Example*	*SR*	*Observed*
pol	... C [function p ...	[natə**t** **p**ɔɮ]	p	3
			w	65 (95.6%)
	... V [function p ...	[ɔrəɮt͡sʰ**ɔ** **p**ɔɮ]	p	2
			w	22 (91.7%)
	... [+nasal] [p ...	[ju**m** **p**ɔɮ]	p	46
			w	25 (35.2%)
pol-	... C [function p ...	[t͡ʃɪɮ**t** **p**ɔɮsən]	p	31
			w	36 (53.7%)
	... V [function p ...	[xit͡sʰ**u** **p**ɔɮt͡ʃʰəxsən]	p	6
			w	18 (75.0%)
	... [+nasal] [p ...	[xu**n** **p**ɔɮəxig]	p	18
			w	1 (5.3%)
pai	... C [function p ...	[pɔttə**g** **p**aisən]	p	58
			w	30 (34.1%)
	... V [function p ...	[tʊtʰ**ʊ** **p**aiɢa]	p	10
			w	53 (84.1%)
	... [+nasal] [p ...	[ʃit͡ʃəxsə**n** **p**aisən]	p	75
			w	6 (7.4%)

pi	... C [function p ...	[pajərla**t pi**]	p	131
			w	5 (3.6%)
	... V [function p ...	[ət**ɔ pi**]	p	12
			w	2 (14.3%)
	... [+nasal] [p ...	[tɛgsə**n pi**]	p	41
			w	0 (0%)

Note that while every function word has its own preference for lenition or not, every function word is still affected by the phonological environment. That is, for all function words, the proportion of lenition after vowels is greater than elsewhere. Likewise, after nasals, the proportion of lenition is always lower. This can be seen clearly in Figure 3, which shows the proportion of lenition in each phonological environment for each function word. The light grey bars indicate the proportion of lenition after vowels, and are highest for every function word except *pol* where rates of lenition in the elsewhere context and after vowels are near ceiling. Likewise, the medium-grey bars, which indicate proportion lenition after nasals, are lowest for all function words.

Figure 3: Lenition rates in K5: intersecting effects of function word and phonological environment



What we see here in K5 is another, more complex, case of interacting pressures in the grammar. It is not enough to label some function words as exceptions. It would be tempting, for instance, to label *pi* as exceptionally non-leniting. But even *pi* lenites the most (percentage-wise) after vowels, and the least after nasals. Both individual lexical preferences and grammatical pressures jointly affect the likelihood of lenition.

To model this in MaxEnt, Fuller (2024) proposes lexically-indexed copies (Pater 2010) of the IDENT(sonorant) constraint as follows:

(30) *Lexically-indexed* IDENT *constraints*

a. IDENT(son)$_{pol}$ Output correspondents of an input [αsonorant] segment in the lexical item *pol* are also [αsonorant].

b. IDENT(son)$_{pai}$ Output correspondents of an input [αsonorant] segment in the lexical item *pai* are also [αsonorant].

c. IDENT(son)$_{pol-}$ Output correspondents of an input [αsonorant] segment in the lexical item *pol-* are also [αsonorant].

d. IDENT(son)$_{pi}$ Output correspondents of an input [αsonorant] segment in the lexical item *pi* are also [αsonorant].

These lexically-indexed constraints act just like their general counterpart IDENT(sonorant), but only assign violations within the lexeme to which they are indexed.

Fuller uses one lexically-specific version of IDENT(sonorant) for each function word in her data, each with a different weight. Because they are all versions of IDENT(sonorant), they will all conflict with *p. Their weight relative to *p reflects that function word's overall propensity for lenition.

The tableau in (31) shows the lexically-specific constraint analysis for K5. The four IDENT(sonorant) constraints assign violations identically, but apply to different inputs, only assigning violations when their indexed lexical item is present in the input. (For convenience we leave out general IDENT(sonorant).) The lexically-indexed IDENT(sonorant) constraints interact with the general phonological constraints from our K4 analysis: *p, *N[+cont], and POSTVOCALICLENITE.

(31) *Tableau for K5*

				*N[+cont]	*p	PostVocLenite	Ident(son)$_{pol}$	Ident(son)$_{pai}$	Ident(son)$_{pol-}$	Ident(son)$_{pi}$			
word	**Context**	**cand.**	**obs.**	2.71	2.23	1.63	0.00	2.59	2.22	5.55	$\mathcal{H}$	p	Obs. p
pol	elsewhere	P	3		1						2.23	.10	.04
		W	65				1				0.0	.90	.96
	after V	P	2		1	1					3.86	.02	.08
		W	22				1				0.0	.98	.92
	after nasal	P	46		1						2.23	.62	.65
		W	25	1			1				2.71	.38	.35
pol-	elsewhere	P	31		1						2.23	.50	.46
		W	36						1		2.22	.50	.54
	after V	P	6		1	1					3.86	.16	.25

		W	18						1		2.22	.84	.75
	after nasal	P	18		1						2.23	.94	.95
		W	1	1					1		4.93	.06	.05
pai	elsewhere	P	58		1						2.23	.59	.66
		W	30					1			2.59	.41	.34
	after V	P	10		1	1					3.86	.22	.16
		W	53					1			2.59	.78	.84
	after nasal	P	75		1						2.23	.96	.93
		W	6	1				1			5.30	.04	.07
pi	elsewhere	P	131		1						2.23	.97	.96
		W	5							1	5.55	.03	.04
	after V	P	12		1	1					3.86	.84	.86
		W	2							1	5.55	.16	.14
	after nasal	P	41		1						2.23	.998	1
		W	0	1						1	8.26	.002	0

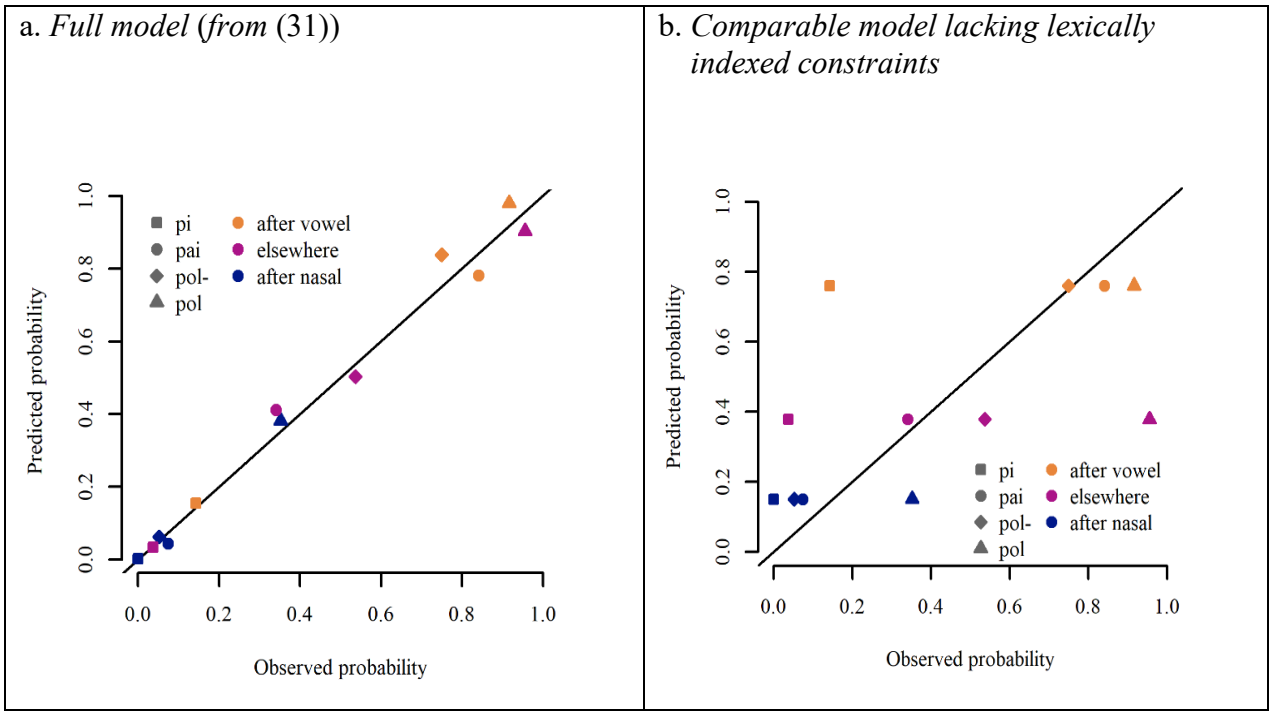
Tableau (31) also contains the weights and predicted probabilities gained from fitting with the Solver using the log likelihood, as detailed in §2.6.4 above. The detailed calculations can be viewed in the **K5.xlsx** spreadsheet in the Supplementary Materials.

The reader may have noticed that the weight of IDENT(son)$_{pol}$ came out as zero. This turns out to be no cause for alarm: since every [w] violates precisely one member of the IDENT(sonorant) family, it turns out that all the matters is the *relative* weights within this family. For discussion, with the relevant math, see §4.4 below.

2.7.1 Assessing the K5 analysis with scatterplots

With K5, we have enough data that it makes sense to visually compare the observed and predicted values in a scatterplot. We compare the model probabilities of each candidate with the fraction of observations that that candidate receives (that is, observed cases divided by total cases for that input.) The left side of Figure 4 shows observed values for the lenition candidates on the x axis, and the corresponding predicted values on the y axis. For purposes of comparison, the graph also contains the line $y = x$. If the model's fit were perfect, every point would be on this line. Indeed, we can see that all the points are very close to the line, indicating a good fit of our model to the data.

Figure 4: Comparing "observed" with predicted values for MaxEnt model of K5



For purposes of comparison, we also give on the right the scatterplot for a model that is similar, but which replaces the four lexically-indexed constraints with one single IDENT(sonorant) constraint (see **K5.xlsx**/K5_noIndexation for weights and computations). For this model, adherence of the points to the $y = x$ line is patently less close; it really did make sense to split up IDENT(sonorant) into morpheme-specific versions. For more on model comparison and the use of scatterplots, see Chapter 5.

2.7.2 K5 vs. real Khalkha

Although our K5 dataset is close to real Khalkha, we briefly point out a few important differences, referring to Fuller (2024). The biggest difference is that in Khalkha, /p/ can actually surface three different ways: as [p], as [w], and as null. This necessitates a richer system of Faithfulness constraints. A second difference is that we have simplified the nature of the phonological perturbers. In real Khalkha, curiously, they seem to form non-natural classes (Mielke 2005). Lenition is dispreferred after nasals [m], [n], and [ŋ] (as in K5) but also after [ɮ] and [w]. The lenition-preferring environment is not postvocalic, but the set consisting of velars and uvulars: [g], [ɢ], [x], plus the diphthong [ui]. With Fuller, we hope that further research will shed light on why these sounds pattern together.

2.8 Visualizing the K5 pattern with shifted sigmoids

At this point we seek to understand the K5 analysis more deeply, particularly how the constraint weights result in the observed patterns of candidate probability. This foreshadows the

theoretical discussion of MaxEnt to be given in Chapter 9. We caution the reader that the discussion of this section gets more technical, and readers will be forgiven if they skip ahead to §2.9. They will, however, miss a theoretical point that might be worth returning to cover later on.

2.8.1 Examining the effects of the weight of one single constraint

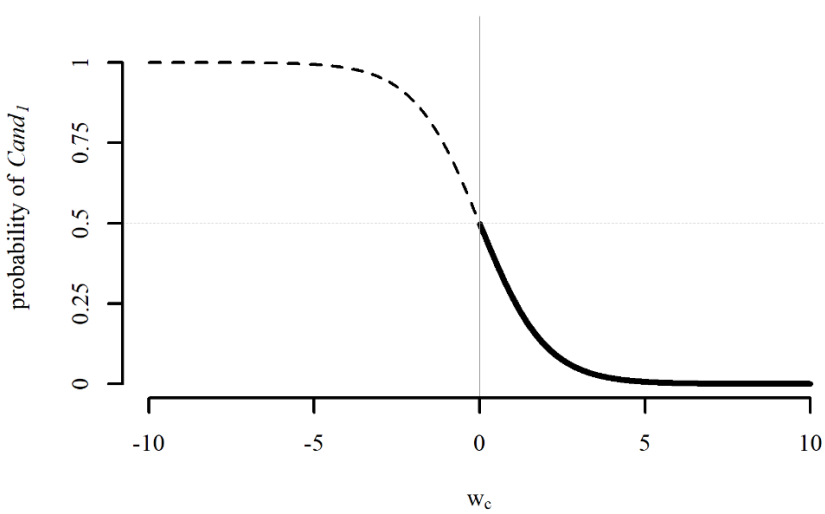
We first examine the relationship between a constraint weight and candidate probability. For this purpose imagine an extremely simple one-constraint grammar in which there are two candidates, one that violates the single constraint ℂ, and one that has no violations at all; hence tableau (32):

(32) *A minimal tableau for illustrating the effect of a constraint weight*

		ℂ **w**
Input	*Cand₁*	
	Cand₂	1

From this tableau, we can use the MaxEnt formula to compute the probability assigned to the candidates — this time, not a single value, but rather a *function* of **w**. Skipping the computation, it emerges that the probability of Cand1 is: $P(\text{Cand1}) = e^{-\mathbf{w}} / (1 + e^{-\mathbf{w}})$. As a function, this can be plotted as a graph, with **w** on the horizontal axis. Seeking full mathematical generality, we allow **w** to go negative, even though negative weights are not commonly used in analytical practice. (Since these values are nonstandard, we plot them with a dotted line.) The graph is shown in Figure 5:

Figure 5: How P(Cand₁) varies with $\mathbf{w}_{\mathbb{C}}$ *in grammar (32)*



This curve is called a **sigmoid**, meaning S-shaped. Looking at Figure 5 in detail, we can observe the following. (a) At $\mathbf{w}_{\mathbb{C}} = 0$, the probability of *Cand₁* is 0.5. This makes sense, because this is the equivalent of a constraintless grammar, and such a grammar should assign 50-50

probability to the two candidates. (b) The sigmoid curve is **monotonic**, in that higher values of $\mathbf{w}_{\mathbb{C}}$ always result in lower values for $P(Cand_1)$. (c) All differences in $\mathbf{w}_{\mathbb{C}}$ will result in differences in $P(Cand_1)$. However, as we increase $\mathbf{w}_{\mathbb{C}}$ going outward from zero, we observe diminishing returns: each additional unit of weight has successively less influence. (d) Probability never actually reaches zero, though the value of $P(Cand_1)$ for high values of $\mathbf{w}_{\mathbb{C}}$ becomes infinitesimally small. (e) In the non-standard region of negative weights, on the left side, the constraint $\mathbb{C}$ acts as a reward, rather than a penalty, and the curve rises going leftward, again with diminishing returns and asymptoting toward 1. (f) The two sides have mirror symmetry; a MaxEnt sigmoid is always symmetrical around the location of 50% probability.

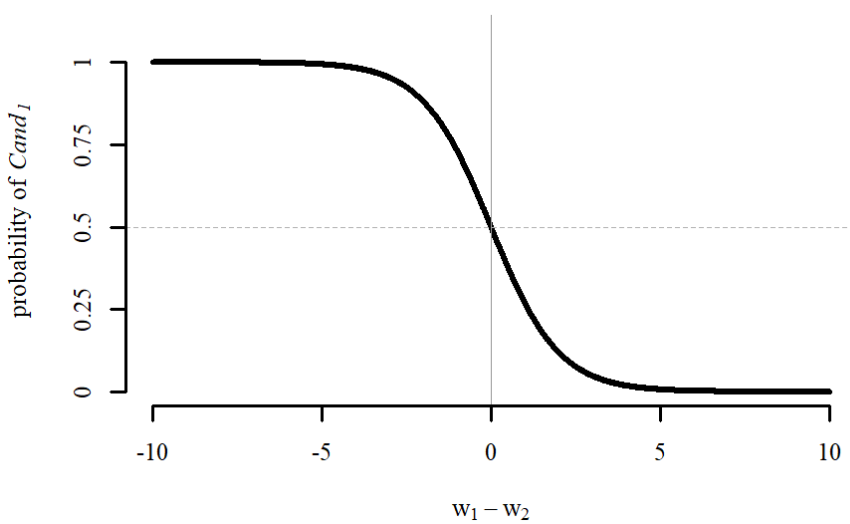
Let us next consider a more realistic example, involving constraint conflict. We suppose that instead of one constraint $\mathbb{C}$ we have two constraints $\mathbb{C}_1$ and $\mathbb{C}_2$, which are opposed to each other. A sample tableau for such a case is in (33).

(33) *A minimal tableau for illustrating the effect of conflicting constraints*

		$\mathbb{C}_1$ $\mathbf{w_1}$	$\mathbb{C}_2$ $\mathbf{w_2}$
Input	*Cand₁*	1	
	Cand₂		1

When we apply the MaxEnt formula to this tableau, it turns out that the only thing that matters is the *difference* between the two weights. We can plot a sigmoid with $\mathbf{w_1} - \mathbf{w_2}$ on the *x* (the equation is $P(Cand_1) = 1 / (1 + e^{w_1 - w_2})$), and it will look just like Figure 5, except that the meaning of the *x*-axis has changed. (We also refrain from dotting the negative part of the line, since it is perfectly possible for $\mathbf{w_1} - \mathbf{w_2}$ to be negative).

Figure 6: How P(*Cand₁*) *varies with* $w_1 - w_2$



Let us return from these abstract cases and ponder the possibility of sigmoid curves in K5. To this end, consider the first input in the tableau (31), where the output probabilities are determined exclusively by the two conflicting constraints *p and IDENT(son)$_{pol}$ — indeed, by the difference in the weights assigned to these constraints. We repeat only the essentials of this tableau in (34),

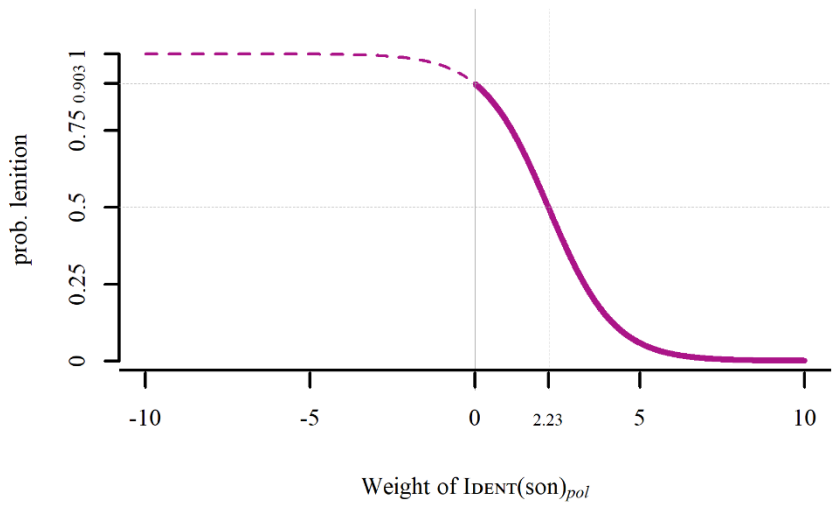
removing the constraints violated by neither candidate. In the full tableau, IDENT(son)$_{pol}$ had a weight of zero, but here we are interested in what would happen if it had different weights; hence the variable *w* in (34).

(34) *Tableau for K5 pol + elsewhere*

word	**context**	**candidate**	*p 2.23	IDENT(son)$_{pol}$ **w**
pol	elsewhere	faithful [p]	1	
		lenited [w]		1

Applying the MaxEnt in the same way as before (yielding p(lenited) = 1 / ($e^{(w-2.23)}$ + 1)), we can plot yet another sigmoid, showing how the probability of the lenited [w] candidate depends on the weight of IDENT(son)$_{pol}$

Figure 7: Sigmoid curve: how P(w) varies when weight of IDENT(son)$_{pol}$ is changed



As can be seen, if the weight of IDENT(son)$_{pol}$ is set very high, lenition is predicted to be very unlikely. If IDENT(son)$_{pol}$ is given a weight identical to that of *p, i.e. 2.23, the rate of lenition is predicted to be 50%; this is indeed the point where the curve crosses the 50% line. As the weight of IDENT(son) approaches zero, the probability of lenition goes up; with $w_{\text{IDENT(son)}} = 0$, it is 90.4%. Lastly, were we to foolishly permit negative weights ("it is bad to be faithful"), we could generate a lenition probability as close to 1 as we wish; this region is shown in the dotted portion of the sigmoid.

In more complicated cases we would actually want to plot the difference in *Harmony* between the two candidates, rather than the different in weights; see for instance §3.3 below. In Figure 7, the two approaches come out the same because there is exactly one violation per candidate.

2.8.2 *The sigmoids arising from perturber constraints*

Let us now consider what sort of sigmoid patterns we get when we introduce perturber constraints into the analysis. Here, we will plot two sigmoids: one for inputs that violate the perturber constraint, and one for inputs that do not. What emerges when we do this — and follows from the MaxEnt math — is that the two sigmoids are *identical and spaced apart by the weight of the perturber constraint*. Following Magri (2025), we will call this configuration **shifted sigmoids**.

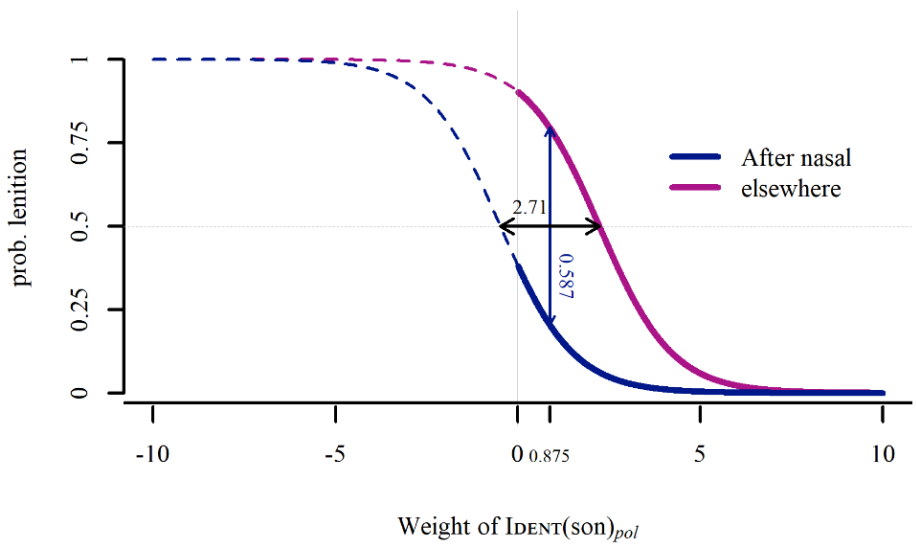
To illustrate shifted sigmoids, we expand our example from (34), this time including *N[+cont], which acts as a perturber constraint, lowering the probability of lenition after nasals.

(35) *Tableau for K5 pol + elsewhere*

Word	**context**	**candidate**	*N[+cont] 2.71	*p 2.23	IDENT(son)$_{pol}$ *w*
Pol	elsewhere	faithful [p]		1	
		lenited [w]			1
Pol	after nasal	faithful [p]		1	
		lenited [w]	1		1

Again, we explore the consequences of varying the weight of IDENT(son)$_{pol}$, but this time plotting a sigmoid for each input; this is shown in Figure 8. Here, the black sigmoid corresponds to inputs with the "elsewhere" environment, and the gray sigmoid corresponds to inputs with the postnasal environments. As shown by the arrow, the two sigmoids are spaced apart on the horizontal axis by 2.71 units, which is, in fact, the weight of *N[+cont].

*Figure 8: An example of shifted sigmoids: identical sigmoids spaced apart by the weight of *N[+cont]*

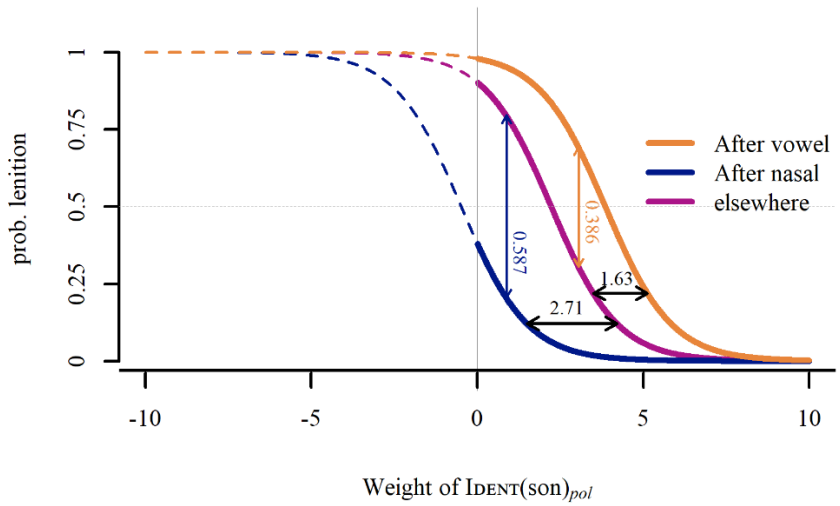


Of course, on the *vertical* axis, the distance between the two sigmoids varies considerably. In the present case, the maximum distance is 0.587, occurring where the weight of IDENT(son)$_{pol}$ is 0.875 (see thin vertical line in Figure 8). The vertical distance approaches zero for ever higher or lower weights of IDENT(son)$_{pol}$. What this means is that the effect of a perturber constraint is *contextually determined*: it is always maximal at the point halfway between where the two sigmoids cross 50%, and is ever smaller going off in either direction.

The contextual variation in the effect of a perturber is a fundamental prediction of MaxEnt, and is repeatedly referred to in analytical work. To reiterate: perturbers are predicted to be strong in their effects when they fall within the middle of the family of related sigmoid curves, but weak on either periphery.

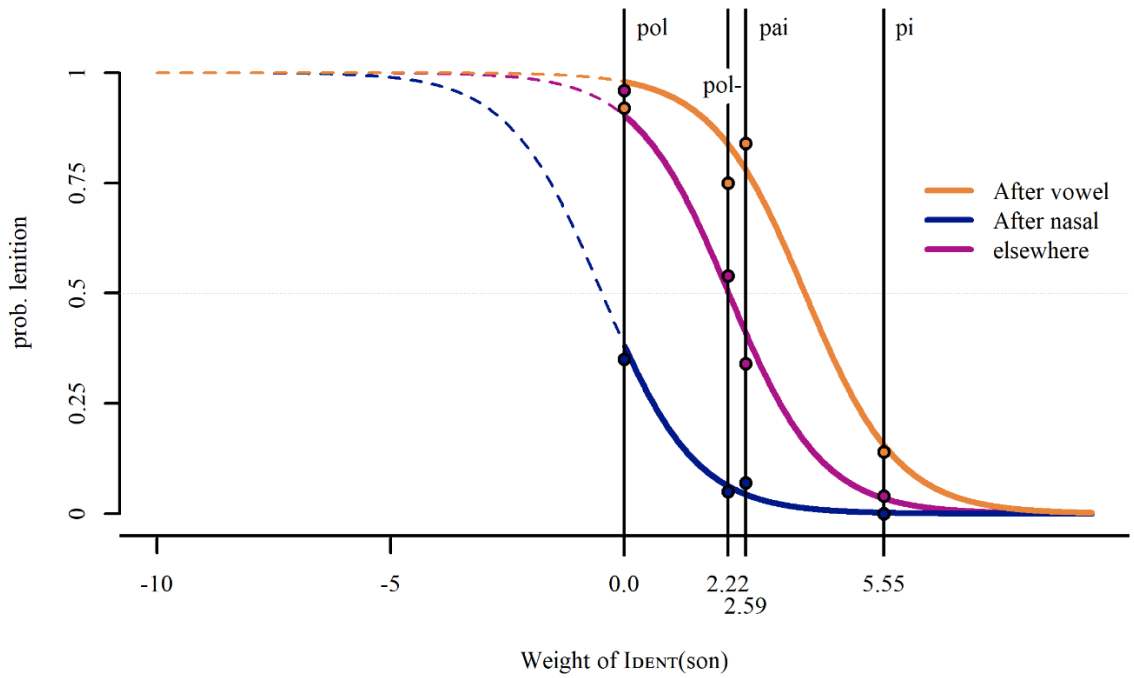
We consider next the case in which there is more than one perturber constraint. Here, there will be three sigmoids, one for the baseline condition and a shifted sigmoid for each perturber. For instance, in Figure 9 we add a second perturber from K5, POSTVOCALICLENITE. This constraint favors lenition postvocalically, so its sigmoid is displaced rightward.

Figure 9: Sigmoid curves for K5 with two perturbers



So far, we have dealt with a hypothetical case in which the weight of IDENT(son)$_{pol}$ is varied freely for illustrative purposes. We next attempt to better understand the actual K5 analysis using the same technique. The sigmoid curves will be identical to those of Figure 9, but we will mark with a vertical line the actual fitted weight for IDENT(son)$_{pol}$, 5.55. We will also include vertical lines marking the weights of the other three morpheme-indexed IDENT(sonorant) constraints, namely IDENT(son)$_{pi}$, IDENT(son)$_{pai}$, and IDENT(son)$_{pol\text{-}}$. The points at which these vertical lines intersect the sigmoid curves represent the predictions of our MaxEnt grammar. The figure also includes dots that show the (imagined) empirical values for the corresponding points in the K5 data. If our fitted model is a good one, then these dots should fall fairly close to the intersection points of the corresponding vertical lines and sigmoids.

Figure 10: Assessing the K5 model using shifted sigmoids



We see that the data points match fairly well with their predicted values. The largest errors are evidently those for *pol-* and *pai* in the postvocalic context, which are 0.09 and −0.06, respectively.

The use of annotated shifted-sigmoid diagrams like Figure 10 provides a useful complement to the use of scatterplots (§2.7.1) in evaluating an analysis, for it indicates how the predictions emerge from the constraints and their weights.

Beyond its role in assessing the accuracy of the analysis, the graph in Figure 10 makes a far more general point, namely the very fact that the data can be fitted with a set of shifted sigmoids. For instance, for the function word *pi*, the high weight of IDENT(son)$_{pi}$ means that lenition is, in general, fairly unlikely for this word. Accordingly, the effects of the perturbers are small. This is seen on the right side of Figure 10, where all the sigmoids are tailing off toward zero. In contrast, for *pai, pol*, and *pol-*, the IDENT(sonorant) constraints bear only modest weights. Thus, we are in the middle stretch of the sigmoid family, where perturbers have substantial effects.

2.9 Very high weights and the approximation of zero

We return for a moment to K4, a language in which some outcomes are obligatory (20). For instance, lenition in K4 is completely blocked in the post-nasal environment. In our MaxEnt analysis, this outcome is attributed to a powerful constraint, *N[+cont], whose best-fit weight is 12.35. For similar reasons, the weight of IDENT(sonorant)$_{content}$ is also very high, 13.37.

These weights must be interpreted with care. First, we note that in fact there is *no* weight assignment that will fit the empirical value of zero exactly. To see this, consider the stage of the MaxEnt calculations given in (16b), where eH is calculated using eH = $e^{-\mathcal{H}}$. No matter how high

is the value of $\mathcal{H}$ (resulting from high constraint weights), $e^{-\mathcal{H}}$ will always be greater than zero; this is in the very nature of the exponential function. So the situation is akin to taking a limit: the higher the weight we assign to *N[+cont] and IDENT(sonorant)$_{content}$, the closer to zero will be the probability assigned to zero-frequency candidates like *[ʃit-t͡ʃʰəx-sən **w**ai-sən] and *[ɔrʊɮ-əx **w**ɔɮəm-t͡ʃʰɔi]. But the value will never actually reach zero; we can only approximate it. The ideal weight for *N[+cont] and IDENT(sonorant)$_{content}$ is, in this sense, infinity.

In the real world, weights are computed by an algorithm that labors to find ever better approximations to the observed distribution. The algorithm will always include one or more parameters that tell it when to stop trying.[13] The values obtained for constraints like *N[+cont] and IDENT(sonorant)$_{content}$ are the values that have been computed up to this stopping point.

There are two lessons here. One is a simple practical one: for some highly-weighted constraints, it is sensible for the analyst not to obsess on the exact weights, even where they differ substantially. All such values are more or less arbitrary stopping points for the search algorithm, the points where its settings lead it to accept the current (tiny) predicted value of zero-frequency candidates as close enough. Perhaps surprisingly, in MaxEnt, the difference in weights between (say) 20 and 50 can be very unimportant, whereas the difference between 1 and 2 might be quite important. We discuss this issue in greater depth in §4.6.

A second lesson goes a bit deeper. Anyone who uses MaxEnt must be prepared to accept an infinitesimally small number as the practical equivalent of zero. For instance, our K4 grammar assigns the probability 3.2×10^{-7} to the bad candidate *[ɔrʊɮ-əx **w**ɔɮəm-t͡ʃʰɔi] and 8.9×10^{-7} to the bad candidate *[ʃit-t͡ʃʰəx-sən **w**ai-sən]; these were shown earlier in (18) for convenience as "≈ 0". The difference between these numbers and zero is smaller than the measurement error for almost all data sets.

Lastly, we point out the class of constraints for which the "infinite weight" syndrome arises: they are the constraints that *prefer a winner*, but never *prefer a loser*, in the terminology of Prince (2003b). In classical Optimality Theory, such constraints are the ones that can always safely be ranked at the top of the grammar. In MaxEnt, they have idealized infinite weights.

2.10 Some Excel tips

Since this is a textbook, we offer a brief section on how one can use Excel more quickly and reliably to carry out MaxEnt calculations.

2.10.1 Avoiding weight crashes

The fact that in some cases a constraint's ideal weight is infinite can raise practical problems in Excel. The Solver's weight-fitting algorithm will halt after a certain amount of computation, often landing on a weight that is high enough to give a good fit to the data. However, in some cases the Solver will fit a constraint weight which predicts probabilities so low that Excel cannot represent them. In this case, the algorithm will return an error message, and cells which cannot

[13] To alter the stopping parameter in the Excel Solver, click Options, GRG Nonlinear, and change the value in the box labeled Convergence.

be represented will display #NUM! instead of actual values. A very simple spreadsheet that creates such a crash is given in (36).

(36) *A MaxEnt spreadsheet that creates an error in Excel*

		Freq.	C1						
			w = 708.3	H	eH	Z	p	ln p	LL
Input1	$Cand_1$	1	0	0	1	~1	~1	~0	~0
	$Cand_2$	0	1	708.3			2.45×10^{-308}	−708.3	

C:\Dropbox\AR\MaxEntBook\PrepareSims\Chapter4

The hand-entered weight in (36), 708.3, is about as high as it can go. If we increase it to 708.4, the tiny probability for $Cand_2$ gets rounded down by Excel to zero. This yields a #NUM! error message in the ln p column, since zero has no logarithm.

We discuss a principled way for dealing with this problem in §4.6 below, but here give an ad hoc solution that does not require much additional calculation. Specifically, it is possible in the Solver simply to impose a hard upper limit on weights. A maximum of 50 is often sufficient to produce an accurate analysis while avoiding crashes. To implement such a limit, do the following:

(37) *An ad hoc procedure for limiting weights*

- In the Solver window, click "Add" to the right of the box labelled "Subject to the Constraints".
- In the left hand field, select the cells which contain the weights.
- Select "<=" for the middle drop-down menu, and type "50" in the right hand field.

The Solver will then give the best-fit answer available, subject to this weight limit: weights that otherwise would be unmanageably large will be limited to 50. For specific problems you may find that an upper limit greater than 50 is necessary, or a limit lower than 50 is sufficient.

2.10.2 Another method for calculating Z

A common source of error in a MaxEnt spreadsheet, often hard to detect, arises in the calculation of the Z values. These must be obtained by summing the eH values (cf. (16b)) for *exactly* the set of candidates for a particular input. Where each input has the same number of candidates, it is not hard to avoid error, but if the inputs can have different numbers of candidates it is easy for a slip of the mouse to create an error. Here is a safer, if more verbose method.

First, you must include (perhaps just adding it in) a column in which the input is specified for every row. For instance, for the first two inputs of (23) we need a column like column B in (38).

(38) *A method for greater reliability in calculating Z*

Excel column names

row #	**B**	C	D	...	H	I	J	K	**L**	M
1	**Example**			...	Id(son)	Postvoc Lenite	Harmony	eH	**Z**	p
2		output	count		4.00	2.00				
3	**g poɣən**	P	290				= SUMPRODUCT (E$2:I$2, E3:I3)	=EXP (-J3)	**=SUMIF(--EXACT(B3,B:B) *K:K)**	=K3/L3
4	**g poɣən**	W	60		1		= SUMPRODUCT (E$2:I$2, E4:I4)	=EXP (-J4)	**=SUMIF(--EXACT(B4,B:B) *K:K)**	=K4/L3
5	**ʊ pai-**	P	10			1	= SUMPRODUCT (E$2:I$2, E5:I5)	=EXP (-J5)	**=SUMIF(--EXACT(B5,B:B) *K:K)**	=K5/L5
6	**ʊ pai-**	W	50		1		= SUMPRODUCT (E$2:I$2, E6:I6)	=EXP (-J6)	**=SUMIF(--EXACT(B6,B:B) *K:K)**	=K6/L5
...	...	...	...	...	...	...	...	...	...	...

Once you have a column in place that repeats the inputs, then the reliable calculation of Z can be based on the formula `=SUMIF(--EXACT(A4,A:A)*J:J)`,[14] which can be entered once into the first cell of the Z column (in (38), cell A4), then copied down the entire column. This formula takes the subset of J (the eH column) where A is equal to the value in A4 (or whatever row you are on), even if it is not contiguous, and sums them. We recommend using this method when you have different numbers of candidates for different inputs, making it dangerous to compute all the Z calculations with separate copy-paste moves. It is also helpful if you plan on experimenting with changing the numbers of candidates for each input.

2.10.3 Troubleshooting

Sometimes a first effort to solve a phonological problem yields blatantly poor results, such as a bad fit to the data for some or all inputs or nonsensical constraint weights (such as zero for a patently necessary constraint). We offer some hints for troubleshooting.

The problem may be with the linguistic analysis; for techniques for fixing shortcomings in this area see §5.2. However, the problem may lie with an incorrect Excel implementation. The following checks may be helpful for such troubleshooting, or indeed, you might find it useful to do them before you ever click the Solve button.

(a) Check that all the constraint violations in your spreadsheet have been correctly assigned.

(b) Visual spot-check: Check a few Harmony values at the beginning, middle and end of your spreadsheet by hand, if feasible. The most common error here is that the Harmony column has left out a constraint; D3:G3, for example, when it should be D3:H3. If you set all the weights by hand to the value 1, it becomes easier to do this check.

[14] The simpler expression `=SUMIF(A4,A:A,J:J)` would also work, but we recommend against it because it groups together inputs that differ only in capitalization.

(c) If you did *not* use the method given in §2.10.2, then hand-check your Z values to make sure that they do indeed sum over all the candidates for the same input.

(d) Check each input's probability distribution, to make sure it sums to 1. For small spreadsheets, you can highlight the candidates for the input with your mouse or tracker and check the sum at the bottom of the Excel screen. To be more thorough, you can use the following formula in cell M4 (in the case of (38)), and copy it down the column: `=IF(SUMIF(A:A,A3,L:L)=1, "", "!")`. Then, search for "!" in that column, which will indicate an output probability distribution that did not sum to 1. If you find an error it could mean that, perhaps, your Z scores were not calculated correctly, or perhaps that your exponentiation is off – it may be you didn't include the negative sign.

(e) Finally, go through each of your constraints and manually change the weight. You should see the probabilities in column L change for all inputs in which some candidate violates that constraint. (You can also check if the log likelihood cell changes.)

(f) It is especially helpful to carry out the above spot checks when you add a new input or candidate to your spreadsheet; this is a frequent source of error.

(g) To obtain consistent results when revising your analysis, it is recommended that you reset the weights to zero before launching the Solver. The benefit of this is that you are more likely to obtain a minimum-weight solution, in which useless constraints will stand out because they are assigned a weight of zero (§5.3.3). For a more principled way of finding the minimum-weight solution, see §4.6.

(h) In larger simulations, the termination criterion employed as default by the Solver may not permit the most accurate feasible solution to be found. In our experience, clicking the Solver button a second time often offers modest improvement in log likelihood. It seems unnecessary ever to click a third time.

2.11 More MaxEnt software

We mention two other forms of software that can implement a MaxEnt grammar and find the best-fit weights.

The **MaxEnt Grammar Tool** is available at brucehayes.org/MaxentGrammarTool/. A number of papers in the literature have used this software to implement a MaxEnt grammar. While it works perfectly well, we currently favor the Excel/Solver combination over it, since Excel renders more transparent the work that has been done.

`Maxent.ot` is a software package usable in the well-known statistical analysis system R. For documentation, see Mayer et al. (2024). It requires the user to have some fluency in writing R scripts, and perhaps is thus less suitable for beginning work. A key advantage of `maxent.ot` is that it lets you embed MaxEnt optimization into larger routines, which is useful if you need to optimize many times; indeed we use it here for this purpose in §5.3.6 and §7.5.3. We suspect

Excel can handle slightly larger problems and is a bit faster (i.e., on a single run). We also find it easier to invent new model types (for instance: one component feeding another) using Excel.

2.12 Where to examine published analyses using MaxEnt

We have not tried to form a complete bibliography of research that has made use of the MaxEnt framework, but various sections of this book provide listings that might be useful for purpose of finding model analyses to emulate. See in particular the discussion of frequency matching in §3.5, the discussion of data-gathering in §5.5, the review of phonotactic work in §6.4, the studies with bias models in §7.6, and the studies cited in the broad review (pan-linguistics) of Chapter 8.

Chapter 3: Using MaxEnt to study lexical frequency-matching

An important empirical advance in phonology occurred in the early years of this century when investigators began to study inconsistent patterns in the lexicon: for instance, cases where certain forms undergo a phonological process and others don't. The two key findings were that exceptionality is *statistically patterned*, and that such patterning is internalized by native speakers, as shown by their ability to extend the pattern to novel words. We give a review of the studies that collectively made this research finding in §3.5, after we have covered a particular example. MaxEnt offers grammars that can describe such patterns of lexical exceptionality as well as the ability of the native speakers to internalize them.

3.1 Tagalog nasal substitution

Tagalog is an Austronesian language spoken widely in the Philippines. For background on this language see Schachter and Otanes (1972). The focus here is on a phonological process of Tagalog known as **nasal substitution**; our treatment here is based almost entirely on the research (including MaxEnt analysis) of Kie Zuraw (Zuraw, 2000; 2010; Zuraw and Hayes, 2017).

3.1.1 The basic data pattern

Nasal substitution is a process triggered by the placement before the stem of a prefix that ends in the phoneme /ŋ/. In such cases, the underlying /ŋ/ of the prefix *merges* with the following stem-initial obstruent, producing an output sound that is a single nasal, taking the place of articulation of the vanished obstruent. For example, if the stem-initial segment is /p/, then the output for /ŋ+p/ will be [m], taking the place of articulation of the /p/ but the nasality of the /ŋ/; thus /pa**ŋ-p**igha'tiʔ/ → /pa**m**igha'tiʔ/ 'being in grief'. More generally, the following changes occur: /ŋp, ŋb/ → [m]; /nt, nd, ns/ → [n]; /ŋk, ŋg/ → [ŋ] and also /ŋʔ/ → [ŋ]. In what follows we will focus on the six stops [p, t, k, b, d, g]; omitting the treatment of [s] and [ʔ].[15]

Tagalog nasal substitution is a classic case of exceptionful phonology: although forms that have undergone nasal substitution are abundant, there are also many forms that do *not* display nasal substitution. For these forms, the output differs little from the input; the only thing that happens is a garden-variety place assimilation, as follows: /ŋp/ → [mp], /ŋb/ → [mb]; /ŋt/ → [nt], /ŋd/ → [nd]; and /ŋk, ŋg/ remain unchanged.

Zuraw (2000:19) emphasizes lexical exceptionality by listing pairs in which the same underlying sequence yields different outcomes. In the data below, "RED" stands for an abstract morpheme that triggers CV reduplication. Normally a particular word will have just one acceptable pronunciation (either substituted or non-substituted), though a few words permit variation between the two options.

[15] As a result the counts and analysis will differ in minor ways from the sources consulted.

(39) *Similar forms that undergo, or do not undergo, nasal substitution*

a. [**p**igha ˈtiʔ]	'grief'
/pa**ŋ-p**igha ˈtiʔ/ → [pa**m**igha ˈtiʔ]	'being in grief'
but	
[**p**o ˈʔok]	'district'
/pa**ŋ-p**o ˈʔok/ → [pa**mp**o ˈʔok]	'local'
b. pag-ˈ**t**uloj]	'staying as guest'
/ka-pa**ŋ-t**uloj-an/ → [kapa**n**uluːjan]	'fellow lodger'
but	
[**t**a ˈboj]	'driving forward'
/pa**ŋ-t**a ˈboy/ → [pa**nt**a ˈboj]	'to goad'
c. [**k**am ˈkam]	'usurpation'
/ma-pa**ŋ-k**am ˈkam/ → [mapa**ŋ**am ˈkam]	'rapacious'
but	
[kalis ˈkis]	'scales'
/paŋ-kalis ˈkis/ → [paŋkalis ˈkis]	'tool for removing scales'
d. [mag-**b**i ˈgaj]	'to give'
/ma**ŋ-b**i ˈgaj/ → [ma**m**i ˈgaj]	'to distribute'
but	
[**b**ig ˈkas]	'pronouncing'
/ma**ŋ**-RED-**b**ig ˈkas] → [ma**mb**ibig ˈkas]	'reciter'
e. [**d**a ˈlaŋin]	'prayer'
/ʔi-pa**ŋ-d**alaŋin/ → [ʔipa**n**a ˈlaŋin]	'to pray'
but	
[**d**i ˈnig]	'audible'
/paŋ-**d**i ˈnig/ → [pa**nd**i ˈnig]	'sense of hearing'
f. [**g**in ˈdaj]	'unsteadiness on feet'
/pa**ŋ**-RED-**g**in ˈdaj/ → [pa**ŋ**iŋin ˈdaj] [16]	'unsteadiness on feet'
but	
[ˈ**g**awaj]	'witchcraft'
/ma**ŋ**-RED-ˈ**g**awaj/ → [ma**ŋg**a ˈgawaj]	'witch'

Zuraw discovered these facts partly by working with Tagalog consultants, partly by consulting a dictionary, and partly by gathering her own corpus. For the latter purpose, she used what is now often called a **web-scraper**; i.e. a computer program that automatically searched the Internet for Tagalog-language web pages, and harvested numerous relevant words from them.

[16] It can be seen that the consonant that results from nasal substitution, [ŋ], shows up *both* in the reduplicative prefix and stem-initially; thus [ŋiŋi]. This intriguing interaction of nasal substitution with reduplication has been known since Bloomfield (1917); for one analysis see McCarthy and Prince (1995, §4.3).

Here, we will be working with a corpus of 866 words from the dictionary study; this is a subset of the corpus used by Zuraw and Hayes (2017).[17] Our selection from the corpus and our MaxEnt calculations may be found in the supplementary materials in **TagalogFull.xlsx**.

We begin with some basic data illustrating patterned exceptionality. In (40), we give the counts for nasally-substituted and non-substituted forms, based on the initial consonant of the stem.

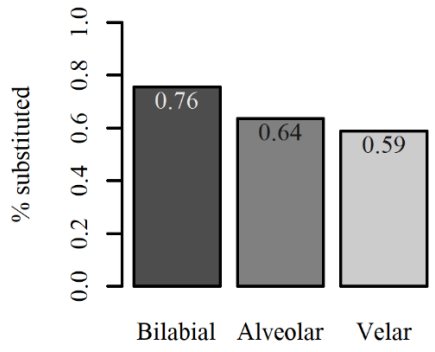
(40) *Tagalog nasal substitution counts by stem-initial consonant*

Stem-initial consonant	*Corpus counts per output type*	
	Not substituted	*Substituted*
p	11	168
t	23.5	129.5
k	18	142
b	86.5	134.5
d	58	12
g	82	1

In the table, fractional values reflect Zuraw's practice of counting forms that have word-specific free variation as 0.5 for each variant.

Within these data, we find two broad, intersecting generalizations. First, the rate of nasal substitution depends on *place of articulation* in a straightforward way: the "further back in the mouth" the stem-initial consonant is articulated, the less nasal substitution will occur. To show this, we replot the data of (40), suitably grouped and re-expressed as fractions, in Figure 11:

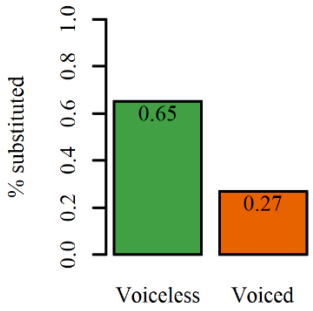
Figure 11: Tagalog nasal substitution: Further-back places have less substitution



A second, stronger generalization is that stems beginning with a voiceless consonant (here [p, t, k]) have a higher substitution rate than those that begin with a voiced consonant (here [b, d, g]), as shown in Figure 12.

[17] The corpus used in Zuraw and Hayes (2017) can be found online at https://muse.jhu.edu/article/670374.

Figure 12: Tagalog nasal substitution: voiceless consonants have more substitution



As we will see, the two factors interact in an additive way, such that [p] (voiceless and bilabial) has the highest substitution rate and [g] (voiced and velar) the lowest. The detailed interaction is shown in Figure 13, which shows that the place effect is quite a bit stronger for voiced consonants:

Figure 13: Tagalog nasal substitution: rates for six consonants plotted separately

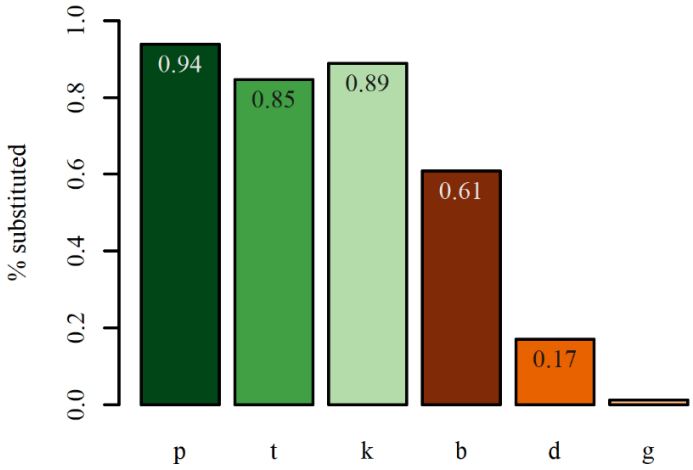


Figure 13 also shows that in this dataset /k/ unexpectly undergoes substitution a bit more often than the fronter consonant /t/. Later on, it will emerge that this is a statistical blip, understandable when we analyze more deeply; see Figure 15.

3.2 An initial MaxEnt analysis

The premise of our analysis is that patterns like the one shown are part of the grammatical knowledge internalized by Tagalog speakers, and that it is appropriate to use MaxEnt to set up a grammar that predicts these patterns. We turn later (§3.4) to the question of whether these patterns are indeed internalized by native speakers.

To start, we will need a constraint that triggers nasal substitution. We follow Zuraw and Hayes (2017:502) in offering a straightforwardly descriptive constraint:

(41) NASSUB: Assess one violation for any *nasal* + *obstruent* sequence, where + is a morpheme boundary within a word.

The tendency of voiceless consonants to take substitution more often than voiceless ones can plausibly be attributed to a widely documented constraint (Hayes and Stivers, 2000; Pater, 2004) that disfavors voicelessness after a nasal:

(42) *NC̥: Assess one violation for every sequence of a nasal and a voiceless obstruent.

In other languages, the same constraint is responsible for post-nasal voicing, pre-voiceless denasalization, or deletion of nasals (Pater 2004). For Tagalog, it will act as a perturber constraint, boosting the nasal substitution rate for stems with /p, t, k/, since in these cases retaining the prefix-final nasal would create a violation, as in [pa**mp**oˈʔok] in (39a).

As for place of articulation effects, we appeal first to a ban on syllables (or stems) that begin with [ŋ]; see for instance English, where there are no words like *[ŋɑp]. The WALS database (Anderson, 2013) suggests that this is a common pattern. Under this approach, forms like (39c) [ma-pa-**ŋ**amˈkam] are disfavored because the surface form has a stem-initial [ŋ]. The difference between alveolars and labials is harder to make sense of; for a possible explanation see Zuraw (2010). For simplicity we posit a constraint against initial [n]. The two constraints are given in (43) below.[18]

(43) *Constraints governing initial nasals*

a. *[$_{\text{stem}}$ ŋ: Penalize candidates where the stem begins with [ŋ].
b. *[$_{\text{stem}}$ n: Penalize candidates where the stem begins with [n].

As for a candidate set, in the interest of brevity we will provide only the two candidates that actually compete in real Tagalog, namely a nasally-substituted one (like /paŋ-poˈʔok/ → [pa**m**oˈʔok]) and a non-substituted one (like [pa**mp**oˈʔok]). A fuller analysis would include fully-Faithful candidates like [pa**ŋp**oˈʔok] (ruled out by a highly-weighted constraint banning non-homorganic nasal clusters), the deletion candidates [pa**ŋ**oˈʔok] and [pa**p**oˈʔok] (ruled out by highly-weighted MAX), and others.

The candidates, constraints, and violations needed so far are given in **TagalogSimple.xlsx**, which looks like this:

[18] The reader may have noticed that we have not provided any Faithfulness constraint violated by the substituted candidate. We will address this lacuna later on (§3.2.1, 4.4.2).

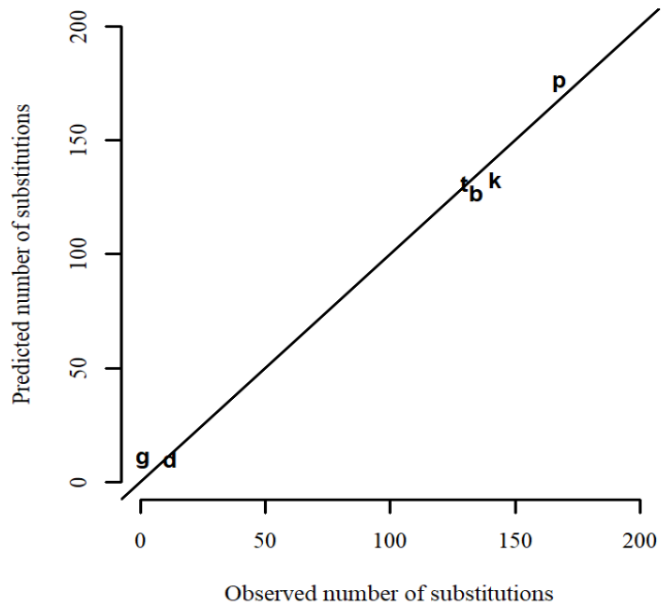
(44) *A MaxEnt tableau spreadsheet for Tagalog nasal substitution (simple version)*

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Consonant**	**Candidate**		NAS SUB	*NC̥	*[n	*[ŋ						
2			**freq**	**0.30**	**3.57**	**2.07**	**2.28**	H	eH	Z	p	ln p	LL
3	p	mam[pX	11	1	1			3.87	0.02	1.02	0.02	-3.89	**-363.71**
4		ma[mX	168					0.00	1.00	1.02	0.98	-0.02	
5	t	man[tX	23.5	1	1			3.87	0.02	0.15	0.14	-1.95	
6		ma[nX	129.5			1		2.07	0.13	0.15	0.86	-0.15	
7	k	maŋ[kX	18	1	1			3.87	0.02	0.12	0.17	-1.78	
8		ma[ŋX	142				1	2.28	0.10	0.12	0.83	-0.19	
9	b	mam[bX	86.5	1				0.30	0.74	1.74	0.42	-0.86	
10		ma[mX	134.5					0.00	1.00	1.74	0.58	-0.55	
11	d	man[dX	58	1				0.30	0.74	0.86	0.85	-0.16	
12		ma[nX	12			1		2.07	0.13	0.86	0.15	-1.92	
13	g	maŋ[gX	82	1				0.30	0.74	0.84	0.88	-0.13	
14		ma[ŋX	1				1	2.28	0.10	0.84	0.12	-2.11	

The weights of the analysis were found using the same method employed in Chapter 2: we used the Excel Solver to maximize log likelihood (cell M3) by adjusting the constraint weights (cells D2-G2). These weights seem intuitively sensible: the fact that *[n has a positive weight reflects the lesser tendency of alveolars to undergo nasal substitution than bilabials; and the still greater weight of *[ŋ captures the fact that velars undergo nasal substitution least often (see Figure 11). The powerful effect of consonant voicing (Figure 12) is reflected in the high weight (3.57) of *NC̥. We also see that the markedness constraint NASSUB plays only a modest role; evidently a good deal of its work is done by *NC̥, whose violations overlap with it.

The fit of the analysis to the data comes out fairly well, as we can see with following scatterplot.

Figure 14: Scatterplot: Observed vs. Predicted counts for the simple Tagalog analysis



In Figure 14, we have chosen to display predicted counts, obtained by multiplying the probabilities for each candidate by the count of forms derived from that input; for further discussion of this method see §5.2.2.

3.2.1 Distinguishing different prefixes

Tagalog resembles Khalkha (§2.7) in that the various affixes that can trigger phonology differ in their propensity to do so. Here, there will be seven affixes under scrutiny; recall that RED stands for an abstract reduplication trigger.

(45) *The Tagalog prefixes examined*

a.	/maŋ-$_{other}$/	non-adversative-verbs
b.	/paŋ-RED-/	mainly gerunds
c.	/maŋ-RED-/	professional or habitual nouns
d.	/maŋ-$_{adv}$/	adversative verbs
e.	/paŋ-$_{noun}$/	various nominalizations
f.	/paŋ-$_{res}$/	reservational adjectives
g.	/mag-kaŋ-RED-/	verbs of accidental result

To illustrate the differences, we return to the same dataset analyzed in the previous section, only this time breaking down the data further according to what prefix is present.

(46) *Data for the full Tagalog simulation*

It can be seen that the prefix complex /mag-kaŋ-RED-/ does not induce nasal substitution at all (Zuraw 2000:119). While the published studies do not include quantitative data for this prefix complex, we have included it in the interest of illustrating how a *non*-variation pattern could be derived in MaxEnt. To fit it in with the rest of the analysis, we adopt synthetic counts that seem reasonable; i.e. 20 to 0 non-substitution for each initial consonant.

Adopting counts based on both stem-initial consonant and prefix, we now have a richer data set embodying independent, intersecting principles: the inputs now cross-classify the six

Consonant	*Prefix*	*sub/unsub*	*Consonant*	*Prefix*	*sub/unsub*
	maŋ-$_{other}$	65/0		maŋ-$_{other}$	66/6
	paŋ-RED-	39/0		paŋ-$_{other}$	29.5/5.5
	maŋ-$_{adv}$	11/0		maŋ-$_{adv}$	6/6
p	maŋ-RED-	18/0	b	maŋ-RED-	16.5/21.5
	paŋ-$_{noun}$	30.5/6.5		paŋ-$_{noun}$	15/30
	paŋ-$_{res}$	4.5/4.5		paŋ-$_{res}$	1.5/17.5
	mag-kaŋ-RED-	0/20		mag-kaŋ-RED-	0/20
	maŋ-$_{other}$	39/0		maŋ-$_{other}$	7/3
	paŋ-RED-	22/0		paŋ-RED-	3/7
	maŋ-$_{adv}$	12/0		maŋ-$_{adv}$	0/12
t	maŋ-RED-	19/0	d	maŋ-RED-	1/12
	paŋ-$_{noun}$	34/17		paŋ-$_{noun}$	1/17
	paŋ-$_{res}$	3.5/6.5		paŋ-$_{res}$	0/7
	mag-kaŋ-RED-	0/20		mag-kaŋ-RED-	0/20
	maŋ-$_{other}$	74.5/0.5		maŋ-$_{other}$	0/13
	paŋ-RED-	25/1		paŋ-RED-	1/17
	maŋ-$_{adv}$	11/0		maŋ-$_{adv}$	0/9
k	maŋ-RED-	20/2	g	maŋ-RED-	0/12
	paŋ-$_{noun}$	9.5/9.5		paŋ-$_{noun}$	0/27
	paŋ-$_{res}$	2/5		paŋ-$_{res}$	0/4
	mag-kaŋ-RED-	0/20		mag-kaŋ-RED-	0/20

consonants /p, t, k, b, d, g/ with the seven prefixes of (45).

In our analysis we will deal with output frequency differences among affixes by setting up affix-specific Faithfulness constraints. These come from the UNIFORMITY family proposed by McCarthy and Prince (1995): such constraints militate against surface segments that are the surface reflexes of two distinct input segments, as in Tagalog /ŋp/ → [m]. The fully-explicit notation provided by McCarthy and Prince employs indices to make the changes clear; for instance we can write /p$_1$a$_2$**ŋ$_3$**-**p$_4$**o$_5$ˈʔ$_6$o$_7$k$_8$/ → [p$_1$a$_2$**m$_{34}$**o$_5$ˈʔ$_6$o$_7$k$_8$], where the double subcripting on the surface segment [m] indicates its status as the merged correspondent of /ŋp/. The general statement of UNIFORMITY is as given in (47).

(47) UNIFORMITY: Assess one violation for every surface segment in correspondence with two underlying segments.

For present purposes, we need particular versions of UNIFORMITY applicable to the various prefixes, as shown in (48).

(48) *Affix-specific* UNIFORMITY *constraints for Tagalog*

a. UNIF-/maŋ-$_{other}$/
b. UNIF-/paŋ-RED-/
c. UNIF-/maŋ-$_{adv}$/
d. UNIF-/maŋ-RED-/
e. UNIF-/paŋ-$_{noun}$/
f. UNIF-/paŋ-$_{res}$/
g. UNIF-/mag-kaŋ-RED-/

These are violated if at least one of the two merging segments is found within the prefix in question.

The new constraints of (48) are included in a substantial tableau, too large to print here but downloadable in spreadsheet form as **TagalogFull.xlsx**. We used this spreadsheet to calculate the best-fit weights for the analysis, which were as in (49).

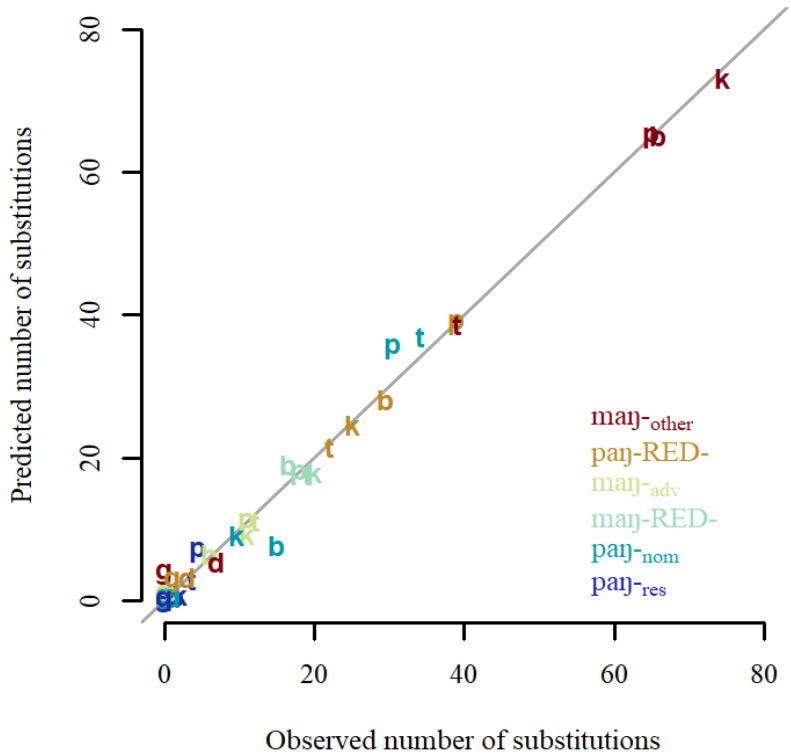
(49) *Constraints and weights for full Tagalog analysis*

NASSUB	2.28
*NC	4.65
*[n	2.10
*[ŋ	3.16
UNIF-/maŋ-$_{other}$/	0[19]
UNIF-/paŋ-RED-/	0.85
UNIF-/maŋ-$_{adv}$/	2.09
UNIF-/maŋ-RED-/	2.27
UNIF-/paŋ-$_{noun}$/	3.85
UNIF-/paŋ-$_{res}$/	5.69
UNIF-/mag-kaŋ-RED-	19.95

The resulting grammar maximizes log likelihood at −248.6, and can be assessed by the following scatterplot of observed vs. predicted counts.

[19] As before in K5, we find that one of the weights comes out as zero; this phenomenon is discussed below in §**Error! Reference source not found.** and §4.4.2.

Figure 15: Scatterplot for the full Tagalog analysis



The fit seems reasonably good, with the evident worst outlier being *paŋ-*$_{nom}$ + *b*, with observed count 15, predicted count 7.7. Examining the errors, it seems there are no obvious constraints that we might want to add in order to increase the accuracy of the analysis. For statistical significance testing of the analysis, see §5.3.4.

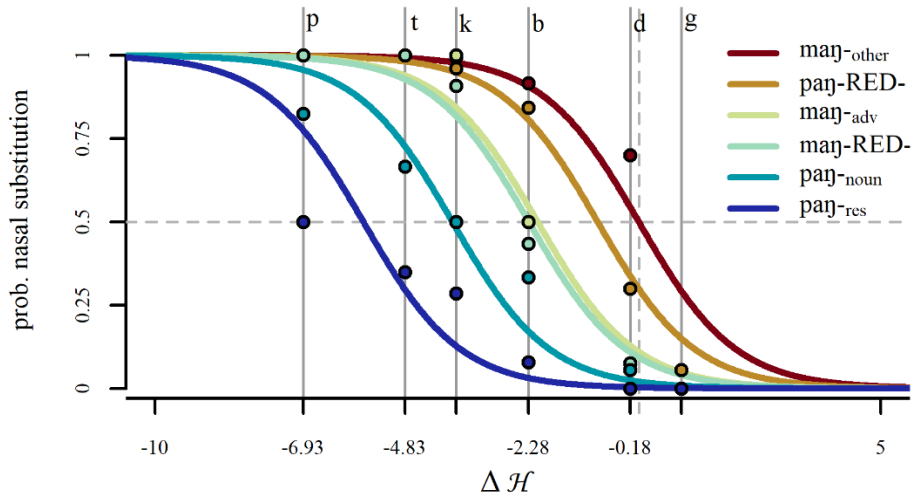
We saw a small anomaly in Figure 13, in that the phoneme /k/ undergoes nasal substitution somewhat more often (88.8%) than /t/ (84.6%), despite the general trend that further back place features undergo substitution less. We can see now that k is not an outlier once we take the behavior of the various individual prefixes into account. In fact, looking back at (46), we can see that /k/ just happens to occur more often than /t/ with *maŋ-*$_{other}$, a prefix that strongly prefers nasal substitution, while /t/ occurs more often than /k/ with *paŋ-*$_{noun}$, which prefers substitution somewhat less. This full model correctly attributes the greater rate of substitution in /k/ forms to the affixes in them, rather than to /k/ itself.

3.3 Shifted sigmoids for Tagalog

Recall that for K5 in §2.7 above, we attempted to diagnose the probability pattern more closely by plotting the data against the family of sigmoid curves generated by the MaxEnt analysis. We do the same here for the Tagalog analysis, taking the opportunity to generalize the procedure somewhat. For K5, the horizontal axis portrayed the weights of a single constraint family, IDENT(sonorant)$_{affix}$. Here, we let the horizontal axis represent simply a difference in harmony between the substituted and the unsubstituted candidate, calculated using all constraints

except the perturbers. In the present case, this will be the summed harmony contributions of NASSUB, *NÇ̥, *[n, and *[ŋ. The perturber constraints will be the six affix-specific UNIFORMITY constraints, leaving out UNIF-/mag-kaŋ-RED since its weight is so high (19.95) that the rising part of the curve would fall falls far to the left of the actual graph, reflecting the fact that no consonant of Tagalog exhibits nasal substitution with this affix. Hence, we are plotting a fairly ambitious set of six shifted sigmoids. As before, the observed data values are included as dots, and model error can be assessed as the vertical displacement of these dots from the intersection of the relevant sigmoid with the vertical line affiliated with the perturber.

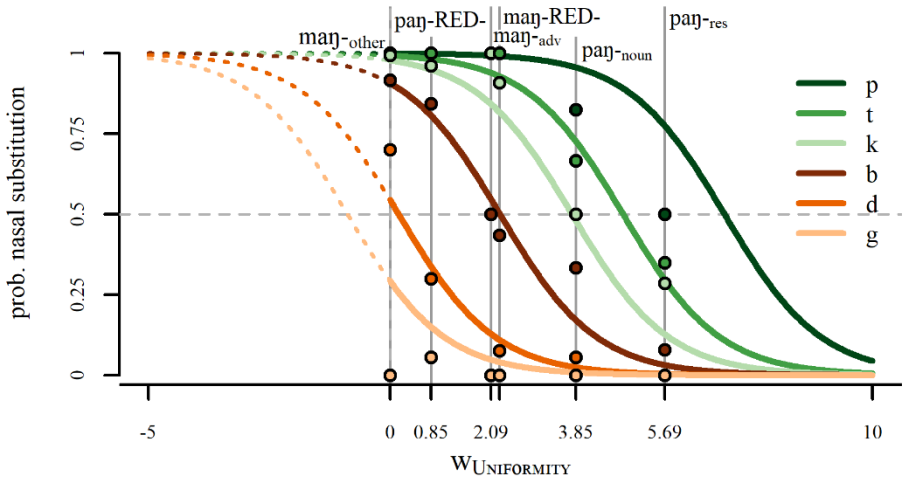
Figure 16: Shifted sigmoids for Tagalog: phonological constraints as baseline



We find that, to a rough extent, the data do seem to be accommodated by a family of parallel sigmoids. As with K5, the arrangement of curves and data point reveals how constraints interact in MaxEnt: more extreme values for baseline harmony cause the effects of prefix to be diminished, i.e. squeezed into a more limited range.

The Tagalog example can also be used to show that the concept of "baseline" vs. "perturber" is really just in the mind of the analyst. We can equally well take a baseline that portrays the weight of the UNIFORMITY constraints, and let the constraints for phonological context act as the perturbers; this yields the shifted sigmoids in Figure 17.

Figure 17: Shifted sigmoids for Tagalog: UNIFORMITY *constraints as baseline*



This display also seems useful to us in that it clearly depicts the most serious outliers in the analysis, namely the overgeneration for [g], /maŋ-$_{other}$/ and the overgeneration for [p], /paŋ-$_{res}$/. We could "fix" these errors by adding constraints that penalized or rewarded these data points in particular, but prefer to retain our analysis, accepting these inaccuracies in exchange for generality and simplicity.

3.4 Frequency-matching grammars and lexical listing

Having developed a grammar that approximately matches the Tagalog lexical frequencies, we must address an important question: should we expect *individual words* to be pronounced varyingly, in probabilities matching the grammar? Empirically, this is not the case: individual words usually have fixed pronunciations, often specific not to the stem, but to the combination of stem and prefix. To give one example (Zuraw 2000:33), the stem /biˈgaj/ 'gift' resists nasal substitution when the prefix is /paŋ-/: [pa**m**-**b**iˈgaj] 'gifts to be distributed'; but undergoes the process when prefixed with /paŋ-/+RED: [pa-**m**i-**m**iˈgaj] 'act of giving away'.

In fact, the probabilistic grammar we have developed describes the aggregate, not the individual case. Zuraw's view (2000, 2010) is that language learners acquire more knowledge than a simple problem-set analysis ever involves: not only do they memorize a large number of polymorphemic lexical entities (such as the cases just mentioned), but they are also aware of the statistical preponderance of the various output patterns.

Zuraw's claim is testable, since it is possible to query native speakers on how they might pronounce imaginary *novel* stems. Her own empirical work (2000, 2010) involved a rating study (modeled on the "wug test" of Berko, 1958), in which Tagalog speakers rated novel forms in which made-up stems either underwent substitution or not. She found that the speakers gave ratings that aligned with the frequency patterns of the Tagalog lexicon. For example, a nonword beginning with /p/ would get a high rating for substitution, a non-word with /g/ would get a low rating for substitution, and a nonword beginning with (say) /b/ or /k/ would get an intermediate rating. Zuraw's finding, which by now have been replicated in many other languages (see next section), will be referred to here as **frequency-matching**.

Frequency-matching may seem somewhat puzzling because it looks redundant: given that speakers already need to memorize individual words (like p*a-**m**i-**m**i* ˈ*gaj* and *pam-bi* ˈ*gaj* above), why should they also bother to represent the general statistical pattern? In fact, there is good reason that language learners should go to trouble of learning these patterns. For instance, a number of prefixes that trigger nasal substitution are relatively productive, and the Tagalog speaker must figure out how to apply the process when a word is new. Beyond this, Tagalog speakers must be able to recognize, and morphologically parse, words they have never heard before; or words from other speakers whose pattern of nasal substitution is different from their own. For these purposes, an accurate internalized "theory of the Tagalog lexicon" would be helpful.

Thus, we make the working assumption that speakers learn a MaxEnt grammar that expresses the statistical patterning of the lexicon — even where the data are idiosyncratic and require memorization. This theory is redundant, and needs to be so, in order match the empirical findings.

It remains to establish just what is the mechanism for memorizing the behavior of individual lexical items. A popular approach is to index the Faithfulness constraints to lexical items; see e.g. Pater, (2000); Becker, (2009); Moore-Cantwell and Pater (2016). Zuraw (2000, 2010) offered a straightforward alternative: the surface forms of the relevant words are memorized. This claim is in line with psycholinguistic work supporting the lexical listing of regular forms (see e.g. Sereno and Jongman, 1997; Baayen et al., 2002; and Zuraw et al., 2021).

3.5 Frequency-matching more broadly

Zuraw's study was one of the first to establish lexical frequency-matching as an ability of native speakers. Since then, numerous studies have found similar results. For in-depth overviews of the literature, see Ernestus (2011), Shaw and Kawahara (2018), Pierrehumbert (2022), Alderete and Finley (2023). Here, we offer a more detailed literature review than elsewhere, since the existence of frequency-matching is the very basis for using a probabilistic framework like MaxEnt.

We discuss six specific areas. In all of them, the research strategy is similar to Zuraw's; i.e. assess a corpus for its patterns and then test the productivity of the patterns with native speakers.

The first cases are like what Zuraw studied, namely frequency matching of patterns of phonological alternation. These include Albright and Hayes (2003) on English past tense formation, Hayes and Londe (2006) as well as Hayes et al. (2009) on Hungarian backness harmony; Jun (2015) on Korean /n/ insertion; Jun and Albright (2017) on Korean consonant alternations in verbal paradigms; Zhang et al. (2006) and Zhang et al. (2009) on Taiwanese tone sandhi; Xu (2025) on Egyptian Arabic vowel patterns; Eddington (1996) on Spanish diphthongization, and Pierrehumbert (2006) on velar softening in English.

A closely related case is allomorphy, i.e. choice among lexically listed alternatives for the same inflectional category. Frequency matching of allomorphy for Hebrew plural suffixes is demonstrated by Becker (2009), for Russian diminutive allomorphs by Gouskova et al. (2015)

and Russian preposition alternations by Linzen et al. (2013), for the forms of the Italian infinitive by Albright (2002), and for the English comparative construction (*more* vs. *-er*) by Gouskova and Ahn (2025).

Another case where the phonological grammar makes a choice is in stress patterns. In many languages stress is unpredictable but there are strong statistical regularities; in choosing a stress pattern for a novel word, experimental participants have been observed to frequency-match these regularities; for English see Guion et al. (2003), Moore-Cantwell (2016, 2020), Olejarczuk and Kapatsinski (2018), and Domahs et al. (2014); similar findings have been obtained by for German and Dutch by Domahs et al., for Portuguese by Garcia (2017), and for Spanish by Eddington (2004).

Another area of frequency-matching in morphophonology is a curious phenomenon first studied by Ernestus and Baayen (2003): in cases of phonological neutralization (e.g., both /rad/ and /rat/ map onto [rat]), speakers show an ability to guess the underlying representation of a form, using phonological cues. These guesses tend to match the statistics of the lexicon, so that URs that are more frequent for a particular phonological context get guessed more often for nonce forms that match that context. We discuss Ernestus and Baayen's study in more detail in §8.2.3. Several studies have found the same effect in other languages: Korean (Kang, 2003; Albright, 2008; Jun, 2010; Jo 2024a), and Turkish (Becker et al., 2011).

In Chapter 2, we studied another form of frequency matching, namely that of free variation in outputs. Here, the frequency matching is carried out by language learners as they learn to reproduce the variation pattern observed in their own language community; thus, the four Khalkha speakers studied by Fuller (2024) ended up producing a similar, suffix- and phonology-specific pattern for /p/ lenition, presumably from exposure to this pattern as produced by older speakers. Similar cases are discussed in Coetzee and Kawahara (2013), Smith and Pater (2020), and Breiss et al. (2024). These results are just some of many, most of which are found in the discipline of classical sociolinguistics: free variation patterns are generally dialect-specific in their details, indicating that young learners are able to attune their idiolects to the frequencies of the ambient language. For discussion and specific examples see Labov (1994:578-580.)

The existence of frequency matching in syntax is supported by Szmrecsanyi et al.'s (2017) study of how the pair of English dative constructions (*Mary gave a book to John* / *Mary gave John a book*) are frequency-matched by speakers of four dialects of English (American, British, Canadian, New Zealand). All four dialects are quantitatively different, implying that the experience of growing up in these different countries leads children to learn to match the local mix of dative constructions. For further discussion on frequency matching in syntax, see §8.6.3.

3.5.1 When is linguistic frequency not matched?

Frequency matching is not always observed. First, language learning appears to be subject to a number of biases, which can lead them to end up with a grammar that deviates from the patterning presented in the lexicon. We discuss a variety of these learning biases in §7.3.

Second, children appear to behave differently from adults with respect to frequency matching behavior generally. For example, Hudson Kam and Newport (2005) have argued for

the view that young children, unlike adults, have a generality bias and tend to match the most frequent pattern only, ignoring less frequent variants. Yang (2016) argues that acquisition is governed by a tolerance threshold, above which children will not frequency-match but will adopt a simple rule, or try to memorize everything; for a critical discussion of this approach, see Jarosz et al. (2025).

In general, we feel that the studies cited above demonstrate that the end-state of language acquisition does involve frequency matching. We note that any frequency-matching grammar necessarily involves substantial detailed knowledge, which plausibly requires many years of experience to develop. At minimum, learners must know enough words to establish a statistical pattern before they can learn it.

Chapter 4: The MaxEnt Math

4.1 Studying the MaxEnt formula

"Probability is nothing but common sense reduced to calculation."

— Pierre-Simon Laplace (1814)

In this section, we adopt the classical Laplacean view of probability, with the goal of providing a closer intuitive interpretion of the MaxEnt formula. De Morgan (1847:172) was following Laplacean views when he wrote: "By degree of probability, we really mean, or ought to mean, degree of belief." Hájek (2023) amends this by requiring that we are referring to *rational* belief; i.e. the degree of belief that would be held by a maximally rational being possessing the relevant facts. The grammar we have hypothesized for Tagalog expresses the native speaker of Tagalog's *rational basis* for holding a particular gradient belief about what will be the outcome for a given input. Because this rational basis is constructed from the violation pattern, it extends to all input forms that share the same violations. In essence, the constraint violations form the evidence to justify the gradient belief.

Let us now take apart the MaxEnt formula, thinking of it as a way of mapping from evidence to rational gradient belief. The goal is to show that at each step, the procedure given by the formula is intuitive and sensible. We repeat the formula below for convenience.

(50) *Maxent formula (verbose version)*

$$p(x)=\frac{e^{-\sum_{i\in Con} w_i\times v_i(x)}}{\sum_{y\in candidates} e^{-\sum_{i\in Con} w_i\times v_i(y)}}$$

*w*here w_i is weight of the *i*th constraint
$v_i(y)$ is the number of times candidate *y* violates the *i*th constraint
Σ_i denotes the sum over all constraints

We can break this formula down into its component parts, and more easily see what each component contributes to the overall behavior of the model.

4.1.1 Weights

Constraint weights appear in the formula as w_i, and express in numerical terms the idea that constraints differ in strength. A constraint with a high weight will generally force candidates which violate it to have low probabilities. A constraint with a low weight will affect the outcome relatively little, and a constraint with a weight of zero will not affect the outcome at all. For more on the interpretation of constraint weights, see §4.4.

4.1.2 Harmony

Weights combine with violation scores to determine each candidate's Harmony. Harmony, calculated in (50) as $\sum_{i \in Con} \mathrm{w}_i \times \mathrm{v}_i(x)$, has a negative relationship to probability: the higher the Harmony, the less probable the candidate.

Since MaxEnt relies on the calculation of Harmony, it is a member of the family of frameworks collectively known as HARMONIC GRAMMAR (Legendre et al., 1990; Smolensky and Legendre, 2006; Potts et al., 2010; Pater, 2016; Boersma and Pater, 2016; Hayes, 2017). The simplest, nonstochastic version of Harmonic Grammar was discussed above in §2.5.1.

The use of Harmony as the basis for the framework, as opposed to the strict ranking principle employed in Optimality Theory, has important consequences for the predictions it makes: the probability predicted for a given candidate is the consequence of *all* of its constraint violations, whereas in Optimality Theory, the choice between two candidates is made solely by the highest ranking constraint that distinguishes them. As Jäger and Rosenbach (2006) put it, theories with Harmony make the prediction of *cumulativity*: multiple violations of a constraint count more than one violation, and different constraints can "gang" together to counter the effects of a more powerful constraints.

In his foundational work on probability theory, Jaynes (2003) suggested that one tenet of sound inductive reasoning is that *no relevant data be ignored*. In this respect, classical Optimality Theory is out on a limb, for indeed its decision procedure (see e.g. Prince, 2003a) frequently does ignore data: specifically, during winner selection, when all losing candidates have been ruled out by higher-ranking constraints, any further constraint violations of the winner are ignored. The discussion of perturber constraints in §2.6.1 constitutes evidence, we feel, that classical OT's strategy of ignoring relevant data is not correct; the very architecture of OT makes it impossible to accommodate perturbers.

Excursus: OT-style decision making in other domains

Classical Optimality Theory finds an echo, perhaps, in the research of Gigerenzer (e.g. Gigerenzer 2001, Gigerenzer and Gaissmaier, 2011) who argues that people often make choices using single salient facts rather than by weighing all the evidence. This model is meant to apply chiefly to cases where data are scarce and speed is essential (Gigerenzer 2001); Gigerenzer and Gaissmaier cite the case of sending a patient to intensive care. In contrast, the decisions involved in learning a grammar are very different: the language learner has a whole childhood's worth of data to go on, with years to ponder and weigh the evidence. Beyond this, existing research suggests that people really do blend a variety of evidence when they make unconscious linguistic choices, as shown by the discussion of perturbers in §2.6.1.

The concept of Harmony permits a terser statement of the MaxEnt formula; instead of (50), we write (51).

(51) *The MaxEnt formula stated in terms of Harmony ($\mathcal{H}$)*

$$p(x) = \frac{e^{-\mathcal{H}(x)}}{\sum_{y \in candidates} e^{-\mathcal{H}(y)}}$$

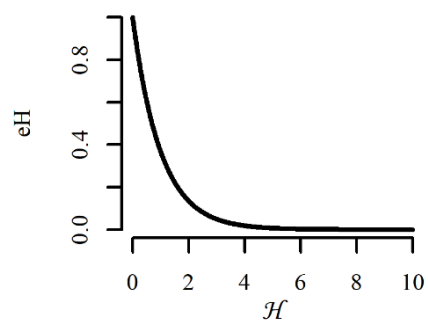
4.1.3 eH, evidence, and nearness to certainty

Having calculated Harmony, we can now calculate eH: $e^{-\mathcal{H}(x)}$. This expression appears as the numerator in the MaxEnt formula as given in (51). We arrive at eH by first negating Harmony, and then exponentiating it using *e* as the base.

The reason for negating the Harmony score is simply to ensure that it will act as a penalty; if Harmony is large, then eH will be small; and smaller eH values ultimately correspond to smaller probabilities.[20]

The use of exponentiation can be made intuitive if we consider how this function "warps" Harmony space. This is demonstrated in Figure 18.

Figure 18: eH values change quickly with low Harmony, slowly with high Harmony



We consider Harmony to be an aggregated measure of evidence: the combined and weighted testimony from all the constraints, that a particular candidate should not be the winner. The consequence of exponentiation is that *added* evidence has effects that are not linear. Specifically, when we have little or no evidence about a candidate, a little more Harmony makes a large difference in eH (and, ultimately, probability). But if we are already close to certainty, then additional Harmony matters very little. For example, in Figure 18, when we move Harmony from 0 to 2, eH changes a great deal; i.e. by more than 0.8. But if we start Harmony off at 8 and move it up to 10, the change in eH is almost imperceptible.

We suggest that this mathematical pattern matches up well to human intuition and experience. As a spur to the imagination, we offer a simple example. Let the novel information be that the celebrated football player Lionel Messi has just pulled a hamstring and cannot play; a lesser player will have to take his place. We place Messi in a match between his national team Argentina and evenly-matched Brazil, and assume that there is only a little time left to play, say, two minutes. The following chart provides, we suspect, a reasonable approximation to normal human intuition, for three scenarios.

[20] Recall that under the sign conventions we adopt (§2.5.2), Harmony is always positive or zero.

(52) *An illustration of the varying effect of evidence on belief*

New evidence: Messi has just pulled a hamstring and cannot play.

Scenario	*Change in* P(*Argentina will defeat Brazil*)
Argentina and Brazil are tied 1-1.	$0.5 \rightarrow 0.4$
Argentina leads Brazil 3-0.	$0.9995 \rightarrow 0.9993$
Argentina trails Brazil 0-3.	$0.0005 \rightarrow 0.0003$

The point is that Messi's injury could hardly matter much when we are already almost certain who will win (it's very rare for three goals to be scored in two minutes, no matter who is playing). When we are not certain about the outcome (game tied), Messi's absence matters much more.

It is not difficult, we feel, to dream up scenarios similar to this one, and we suspect that the pattern seen — and hence the MaxEnt conversion of $\mathcal{H}$ to eH — embodies what we think of as common sense.

The concept of eH permits a still-terser statement of the MaxEnt formula, as in (53).

(53) *The MaxEnt formula stated in terms of eH*

$$\mathrm{p}(x) = \frac{\mathrm{eH}(x)}{\sum_{y \in candidates} \mathrm{eH}(y)}$$

4.1.4 Computing Z and probability

In order to convert eH into a probability, we must ensure that the probabilities for all candidate outputs for a single input sum to 1. This is what the denominator in (50) does: we create a normalizing value Z by summing the eH values across candidates. The crucial consequence of this is that the probability of a candidate depends not just on its own eH, but on that of its competitors; that is, a candidate gets a lower probability when it must compete with strong rivals.

4.1.5 Local summary

In sum, the MaxEnt formula (50) defines a system in which constraints vary in their strength, where the meaning of strength is "ability to lower the predicted probability of violating candidates". The contributions of the various constraints are pooled, reflecting the idea that no relevant evidence should be ignored. The conversion of harmony to eH means that the same bit of novel evidence matters most when the decision is still in doubt, but least when the decision is close to certain. Finally, the division by Z expresses the commonsense notion that an outcome is less probable when it competes with strong alternatives. In sum, MaxEnt is a particular embodiment of the claim made by Laplace, quoted at the start of this chapter.

4.2 Relationship of MaxEnt to classical Optimality Theory

For every grammar formulated in classical Optimality Theory, there exists a close MaxEnt equivalent (Eisner, 2000; Manning, 2003). The trick in finding it is to reexpress all cases of OT constraint domination as differences in weight that are sufficient to assign essentially-zero probabilities to the losing candidates.

To illustrate this, we return to the non-stochastic toy language K1, whose OT tableau was given in (5). We give here an alternative MaxEnt tableau that generates closely equivalent outcomes.

(54) *Re-expressing the OT analysis of (5) in MaxEnt*

	IDENT (son)$_{cont.}$	*N [+cont]	*p	IDENT (son)				
/mʊntəg **p**oɮ-ən/	120	120	30	0	$\mathcal{H}$	eH	Z	p
faithful [p]			1		30	9.36×10^{-14}	≈ 1	9.36×10^{-14}
lenited [w]				1	0	≈ 1	≈ 1	≈ 1
/ʃit-t͡ʃʰəx-sə**n** **p**ai-sən/								
faithful [p]			1		30	9.36×10^{-14}	9.36×10^{-14}	≈ 1
lenited [w]		1		1	120	7.67×10^{-53}	9.36×10^{-14}	8.19×10^{-40}
/ɔrʊɮ-əx **p**ɔɮəm-t͡ʃtʰɔi/								
faithful [p]	1		1		30	9.36×10^{-14}	9.36×10^{-14}	≈ 1
lenited [w]				1	120	7.67×10^{-53}	9.36×10^{-14}	8.19×10^{-40}

The constraint weights in (54) essentially recapitulate the domination relations portrayed earlier in 2.1: the relation *p ≫ IDENT(sonorant) is expressed by a weight difference in (54) of 30 (= 30 − 0) between these two constraints. The domination relation IDENT(sonorant)$_{content}$ ≫ *p is expressed by a weight difference of 90 (120 − 30) between these two constraints, and the same holds for *N[+cont] > *p. The resulting probabilities for the loser candidates of the OT analysis are all below 10^{-13}, a value that we have decided (§2.9) to accept as the practical equivalent of zero. For more technical discussion of how to reexpress an OT grammar as a weight-based grammar, see Prince (2003a).[21] For another example of an OT grammar rendered in MaxEnt, see §7.9.1.

The upshot of the present discussion is that MaxEnt can be thought of not so much as a replacement for classical OT, but as a probabilistic extension of it; any result derived in classical

[21] Prince's discussion relates to non-stochastic Classical Harmonic Grammar, described above in §2.5.1. To cover MaxEnt, the weight differences Prince calculates must be augmented to cover a large interval corresponding to what we are willing to accept as the equivalent of strict ranking (Eisner 2000).

OT could in principle be derived in MaxEnt as well. Given the great body of empirical material that has been addressed in classical OT, this is important.

4.3 Relationship of MaxEnt to multinomial regression and log-linear models

MaxEnt OT inherits both its name and its math from a class of statistical models, now generally referred to as **log-linear models**, or as **multinomial logistic regression**. As the purpose of this section is to clarify the distinction between the two, we will here be using "MaxEnt OT" to refer to the linguistic model covered in this book, and "log-linear models" to refer to the class of statistical models. For training in the use of multinomial logistic regression, we refer the reader to Field (2026:ch. 19) and Jurafsky and Martin (2025:ch. 5).

The point we wish to make is that despite having identical math, MaxEnt OT and log-linear statistical models are very different in their intellectual goals. Statistical models are a general tool used to assist scientific inquiry; they are designed to assess a set of predictors for how much of the variation in a given data set they can account for. Because the goal of a statistical analysis is to understand a specific dataset as thoroughly as possible, it is most informative when it is created bespoke for the specific dataset. In contrast, MaxEnt OT aspires to model how human minds represent phonological generalizations. To be useful for this purpose, it must do more than model a single dataset; it must also be responsible to phonological typology.

We note in passing that the close relationship between MaxEnt OT and log-linear models means that statistical methods developed for log-linear models can be adapted to MaxEnt OT models. Examples of this include the use of random effects to model lexical exceptionality (Zymet 2018; Hughto et al. 2019; Linzen et al. 2013) and lexical classes (Shih 2018), the use of regularization priors to model learning biases (Wilson 2006 and Chapter 7 below); and the calculation of confidence intervals on weights (Storme 2026).

In the sections below, we discuss a few cases in which best practices for phonological modeling differ from best practices for statistical modeling. We give practical advice for these issues in §4.3.3.

4.3.1 Choosing predictors vs. choosing constraints

When fitting a log-linear model to data, predictors should be chosen so as to cover all factors that could possibly influence the data, including "nuisance" variables such as trial order, the side of the screen on which a button press is made, and so on. Only predictors which logically cannot influence the data can be safely left out of the analysis, such as number of syllables in a study with only two-syllable words.

Constraints in MaxEnt OT are chosen by different criteria: the constraint set embodies a phonological theory, which makes substantive predictions about what properties phonological systems can be sensitive to and hence about what phenomena can occur in the world's languages. A particularly effective research method in OT has been to compute the **factorial typology** of the constraint set; i.e., the set of outcomes obtained under all possible constraint rankings; see e.g. Kager (1999, §1.7); and for empirical examples Gordon (2002) and Kaun (2004). It is possible to consider the typological implications of a constraint set in MaxEnt as well, but the

structure of that analysis is somewhat different; see Anttila and Magri (2018). In addition to substantive claims about exactly what is in the universal constraint set, OT theorists also sometimes address limits on the formal character of the constraints; see e.g. Eisner (1997b) and McCarthy (2003).

Another difference is that most regression models rely on an assumption of independence: that predictors are not highly correlated with each other. For discussion of the problems that arise when a statistical analysis is carried out with correlated predictors, see Çetinkaya-Rundel & Hardin (2024, §25.2). In contrast, MaxEnt OT analysis often employs constraints whose violation profiles are correlated. For example, there may be good theoretical reasons to include in the same analysis both a constraint penalizing all voiced obstruents (*VOICEDOBS), and a constraint penalizing just the phone [g] (*g) even though they are strongly correlated (since every time *g is violated, so is *VOICEDOBS). Such choices are made for the reason given above, namely that constraints are part of a general theory.

4.3.2 Choice of candidate set

Part of the basic training given to OT analysts (see e.g. McCarthy 2008) is in the choice of a candidate set sufficient to provide a true test of the analysis. The constraint set and weights in (MaxEnt) OT are responsible not just for predicting probabilities over possible outcomes, but also for explaining why each input does *not* surface in any impossible ways. Beyond this, OT analysts must also be skilled in selecting the appropriate *inputs* for the analysis; often an analysis can be shown to fail because it makes wrong predictions for inputs not considered.

In contrast, statistical models are designed to fit actually-observed data; for any given experiment, the impossibility of cases not tested is simply assumed in advance, as part of the experimental design (though in principle it could be checked in subsequent experiments).

Excursus: The infinite candidate set in MaxEnt

One of the boldest innovations of OT was to suppose that outputs are selected from an infinite candidate set, i.e. GEN. Since human minds are obviously finite, this principle represents a strong idealization; for discussion see Prince and Smolensky (1993:215-216). These authors presciently noted that "in practice, computationalists have always proved resourceful," and indeed, subsequent research soon demonstrated that it is possible to set up implemented GEN functions that are infinite yet reliably searchable in finite time. The key idea is to represent GEN not as a list but as an abstract schema, often implemented with a finite state machine, as in Ellison (1994), Eisner (1997a) and Riggle (2009). Such work has not only provided reassurance on the tractability of infinite candidate sets, but also improved the reliability of hand analysis (Karttunen 2006). The approach of abstract schemata has been carried over to MaxEnt work, e.g. in Hayes and Wilson (2008).

In MaxEnt, infinite GEN raises an even more alarming prospect. We know that every candidate receives a positive value for eH; these values are summed ((16c)) to obtain the denominator Z. With infinite GEN, we have to worry about Z coming out infinite. This would in fact be catastrophic, since any number divided by infinity is zero, which means that all

predicted probabilities would be zero, and the grammar would not distinguish between candidates.

Fortunately, it turns out that not all constraint sets give rise to this outcome. The key requirement is set forth by Daland (2015): roughly, there must be at least one constraint whose violations scale with the length of the candidates. *STRUC (Prince and Smolensky 1993:25) and DEP (McCarthy and Prince 1995:264) are constraints of this sort. Suitably weighted, such constraints guarantee that the eH values assigned to a set of candidates of ever-increasing length will follow a descending geometric series. If the weights are positive, then the series will sum to a finite number. For example, if $w_{*\text{STRUC}} = -\ln(0.9) \approx 0.109$, then each candidate in the series [a], [aa], [aaa], … will receive an eH that is 0.9 smaller than its predecessor. The sum that results ($0.9 + 0.9^2 + 0.9^3 + \ldots$) is finite (in fact, it equals 9). We suggest that the possibility of infinite Z is, at least logically speaking, a peril of which practioners should be aware, but in light of Daland's result, ordinary analytical practice probably remains safe.

4.3.3 Using both statistical models and MaxEnt models together in research

The purpose of a log-linear statistical model is not to be a theory of phonology, but rather to assess a set of predictors for how much of the variation in a given data set they can account for. In contrast, a MaxEnt OT model is a model of the grammar internalized by the native speaker, as diagnosed by their outputs, intuitions, and experimental behaviors. Thus, if a program of investigation includes multiple experiments or other probes, ideally they should all be modeled with a single MaxEnt grammar, though each is subjected to it own statistical analysis. Indeed, even if one is doing just one experiment, it may make sense to provide both a statistical analysis of the experimental result and a hypothesized MaxEnt grammar describing the speakers competence; for paper that have done this see Hayes et al. (2009), and Moore-Cantwell (2020).

To sum up this section, we have suggested that despite the identical math used, statistical anlaysis of experiments and MaxEnt modeling of grammars are distinct enterprises. A statistical analysis can check the influence of nuisance variables and can use ad hoc reformulations of the phonological principles to avoid correlation between predictors. A MaxEnt theoretical model can address all plausible inputs and candidates (including those not tested), can integrate the results of several experiments, and can attempt to make sense of the data in terms of a particular linguistic theory that covers what a constraint can be like. This both tests the theory and renders the research responsible to typology. When we subject experimental results to statistical testing, we are seeking reassurance that they can be used as reliable building blocks for the construction of a phonological model.

4.4 Interpreting constraint weights

4.4.1 How weights determine probability: log odds in MaxEnt

In §4.1.1, we gave the simple intuitive meaning of weights: the higher a constraint's weight, the greater its effect in lowering the probability of candidates that violate it. However, constraint weights actually have a very specific relationship to the probabilities they predict. We illustrate

this relationship here, with the intent that the reader can use it to develop an intuitive sense of what the real-number value of a constraint weight means.

Consider a very simple tableau, with two candidates and one constraint ℂ.

(55) *MaxEnt tableau: one constraint, two candidates*

		ℂ $w_{\mathbb{C}}$	$\mathcal{H}$	eH	Z	p
Input	*Cand₁*		0	1	$1+e^{-w_{\mathbb{C}}}$	$\frac{1}{1+e^{-w_{\mathbb{C}}}}$
	Cand₂	1	$w_{\mathbb{C}}$	$e^{-w_{\mathbb{C}}}$		$\frac{e^{-w_{\mathbb{C}}}}{1+e^{-w_{\mathbb{C}}}}$

The important thing here is that *Cand₂* incurs exactly one more violation than *Cand₁*. In this case, the ratio of the probabilities of *Cand₁* and *Cand₂* — what a gambler would call the *odds* — is completely determined by the weight of ℂ. We used the MaxEnt formula (computations given in the right columns of (55)) to calculate these odds for various values of $w_{\mathbb{C}}$, shown in (56). As can been seen, the value of the odds increases non-linearly with $w_{\mathbb{C}}$. We find it useful in the intuitive understanding of MaxEnt weights to keep some approximation of this table in our head.

(56) *Probability ratios (odds) of Cand₁ to Cand₂, as determined by* $w_{\mathbb{C}}$, *in scenario* (55)

$w_{\mathbb{C}}$	*Odds*
0	1.00
0.1	1.11
0.2	1.22
0.5	1.65
1	2.72 = *e*
2	7.39
5	148.41
10	22026.47

A weight of 0 yields an odds of 1, or '1 to 1', meaning that *Cand₁* and *Cand₂* are equally likely. A small weight like 0.1 has only a subtle effect, whereas a weight of 10 renders the odds overwhelmingly in favor of *Cand₁*. The probability of *Cand₂* will be empirically indistinguishable from zero at this weight.

The table in (56) is easily generalized: as stated it expresses the difference in harmony resulting from one single constraint, but the same results would hold for *any difference in harmony* (call it $\Delta\mathcal{H}$), whether arising from one constraint or several. In the Excursus box, we give a simple proof that justifies this claim. It tells us that $\Delta\mathcal{H}$ is equal to the log of the odds of two candidates whose Harmonies differ by $\Delta\mathcal{H}$ or in mathematical notation, $\ln(O)=\Delta\mathcal{H}$.

Excursus: The log-odds principle

Here, we show that any two candidates for the same input whose Harmony values differ by the quantity $\Delta\mathcal{H}$ will be assigned a probability ratio O (short for "odds") whose natural log is $\Delta\mathcal{H}$. In the proof below, the term "log odds" is used to mean "log of the odds".

(57) *The log-odds principle*

- Let P(c*and*$_1$) and P(c*and*$_2$) be the probabilities assign by a MaxEnt grammar $\mathbb{G}$ to two candidates for the same input.
- Let O be the odds assigned by $\mathbb{G}$ to these candidates: O = P(*Cand*$_1$)/P(*Cand*$_2$)
- Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be the Harmony values assigned by $\mathbb{G}$ to these candidates.
- Let $\Delta\mathcal{H} = \mathcal{H}_2 - \Delta\mathcal{H}_1$, the difference in harmony between the candidates.

Then:
$\ln(O)=\Delta\mathcal{H}$

In other words, the harmony difference is equal to the log odds.

Proof:

$O=\frac{P(cand_1)}{P(cand_2)}$	Definition of O
$=\frac{e^{-\mathcal{H}_1}/Z}{e^{-\mathcal{H}_2}/Z}$	Plugging in the MaxEnt formula
$=\frac{e^{-\mathcal{H}_1}}{e^{-\mathcal{H}_2}}$	Multiply denominator and numerator by Z
$=\frac{e^{-\mathcal{H}_1}}{e^{-(\mathcal{H}_1+\Delta\mathcal{H})}}$	By definition of $\Delta\mathcal{H}$
$=\frac{e^{-\mathcal{H}_1}}{e^{(-\mathcal{H}_1-\Delta\mathcal{H})}}$	Distribute the negative sign
$=\frac{e^{-\mathcal{H}_1}}{e^{-\mathcal{H}_1}\times e^{-\Delta\mathcal{H}}}$	$x^{a+b}=x^a\times x^b$
$=\frac{1}{e^{-\Delta\mathcal{H}}}$	Divide denominator and numerator by $e^{-\mathcal{H}_1}$
$= e^{\Delta\mathcal{H}}$	$\frac{1}{x^{-a}} = x^a$
and so:	
$O = e^{\Delta\mathcal{H}}$	
$\ln(O)=\ln(e^{\Delta\mathcal{H}})$	Take the natural logarithm of both sides.
$\ln(O)=\Delta\mathcal{H}$	$\ln(e^x) = x$

The log-odds principle has important consequences for MaxEnt analysis. Notably, in many analyses, the weights of individual constraints cannot be fully determined, since all that really matters is the $\Delta\mathcal{H}$ values for each pair of candidates. There are various examples of this kind; a very simple one is the case where *Cand1* violates the two constraints $\mathbb{C}_1$ and $\mathbb{C}_2$, and *Cand2* violates none. Here, any weighting of $\mathbb{C}_1$ and $\mathbb{C}_2$ in which w_1 and w_2 have the same sum will produce the same result; see (58).

(58) *Tableau: Any set of w_1 and w_2 whose sum is the same yields the same result*

		$\mathbb{C}_1$ **w_1**	$\mathbb{C}_2$ **w_2**	$\mathcal{H}$
Input	*Cand1*	1	1	w_1+w_2
	Cand2	0	0	0

Another simple case in which an infinite number of weightings will yield the same result also involves two candidates and two constraints, but with the violations arranged in a diagonal pattern, as shown in (59). For this case we provide a worked-out example, **DifferentWeightsYieldSameOutcome.xlsx**. The violation scheme here is abstract, but is instantiated in analysis over and over by a Markedness constraint and an opposing Faithfulness constraint.

(59) *Two constraints and two candidates with 75/25 probability*

			$\mathbb{C}_1$ **2**	$\mathbb{C}_2$ **0.901**				
		or:	**1.099**	**0**	$\mathcal{H}$	eH	Z	p
Input	*Cand1*	0.75		1	0.90	0.41	0.54	0.75
	Cand2	0.25	1		2	0.14		0.25

We find that if w_1 is held constant at 2, then the weight for $\mathbb{C}_2$ that yields the 75/25 probability difference is 0.901. But as the Log-Odds Principle tells us, *any* pair of weights that differs by the same amount — in this case, 1.099 — will yield the same answer; the reader can easily verify this by changing the weights in the spreadsheet.

Among the pairs of weights that differ by the value 1.099 will be what we listed in the third row of (59), namely $\mathbf{w_2}$= 0 and $\mathbf{w_1}$ = 1.099. This looks perhaps puzzling, because $\mathbb{C}_2$ seems to have "dropped out of the grammar" — by the MaxEnt math it can have no influence on the outcome. We will cover this phenomenon in more detail here, because in our experience, the appearance of surprising zeroes is an ordinary occurrence in MaxEnt analysis — even for constraints that phonologists intuitively feel belong in the grammar. We have already seen this happen in K5 (§2.7) and Tagalog (§3.2.1). In the next section we take a closer look at the Tagalog case.

Before moving on, we note that the effect of a constraint weight on log odds must be interpreted in the context of the rest of the analysis. A constraint $\mathbb{C}$ with weight **w** = 1.099 will

always reduce the probability of a violating candidate relative to a non-violating one by a factor of 3. In the example above, this was a reduction from 0.75 to 0.25. However, in cases where the candidate competition is dominated by other, highly weighted constraints, then the reduction may be very small, say, from 0.000000075 to 0.000000025. Obviously only the first case matters empirically. We have seen such diminishing effectiveness of constraints before; see discussion in §2.8 and §4.1.3.

4.4.2 Zero weights in practice

Having studied the two-constraint case, we can now see the Tagalog case as simply a more elaborate instance of the same thing. Corresponding to $\mathbb{C}_1$ is NASSUB, corresponding to $\mathbb{C}_2$is the *whole family* of UNIFORMITY constraints. We can see this for the particular tableau covering the simplest case, with stem-initial /b/, shown in (60).

(60) *An illustrative tableau for Tagalog stems starting with /b/*

			NASSUB	UNIF-/maŋ-$_{other}$/	UNIF-/paŋ-RED-/	UNIF-/maŋ-$_{adv}$/	UNIF-/maŋ-RED-/	UNIF-/paŋ-$_{noun}$/	UNIF-/paŋ-$_{res}$/
prefix	**cand.**	**obs.**	3.28	0	0.85	2.09	2.27	3.85	5.69
maŋ-$_{other}$	unsub	6	1						
	sub	66		1					
paŋ-RED-	unsub	5.5	1						
	sub	29.5			1				
maŋ-$_{adv}$	unsub	6	1						
	sub	6				1			
maŋ-RED-	unsub	21.5	1						
	sub	16.5					1		
paŋ-$_{noun}$	unsub	30	1						
	sub	15						1	
paŋ-$_{res}$	unsub	17.5	1						
	sub	1.5							1

Crucially, any one candidate with substitution violates precisely one of the six UNIFORMITY constraints. It should be clear that, just as before, we can add an arbitrary amount to NASSUB, and the same amount to every member of UNIFORMITY, and this addition will make no difference to the output probabilities, by the same reasoning — each input, viewed by itself, will have the "diagonal" violation pattern that we saw in the two-constraint example (59). As with the two-constraint example, a weighting algorithm such as the one used in the Excel Solver, which attempts to keep the weights low, will likely assign a zero to the weakest of the UNIFORMITY weights, which happens to be the weight for UNIF-/maŋ-$_{other}$/.

Seeking to understand the zero weight further, we observe that this analysis, like all generative grammar analyses, is intended as a *fragment*, covering but a small area of the vast system internalized by the native speaker. We believe that the zero weights obtained for Tagalog above are indeed the result of an incomplete analysis. For simplicity we deployed an analysis with only two candidates per input. But there are other candidates that violate UNIFORMITY. Consider for instance, a candidate like [ãmigaj] from underlying /ma**ŋ-b**iˈgaj/; where the [ã] corresponds to both the input m and the input a (in the formal notation of McCarthy and Prince (1995), this is /m_1a_1**$ŋ_3$-b_4**iˈgaj/ --> [$ã_{1,2}$**$m_{3,4}$**iˈgaj]).[22] Since the segments /m/ and /a/ have been phonologically merged into [a], and since they belong to the prefix named "maŋ-$_{other}$," this candidate incurs a violation of UNIF-/maŋ-$_{other}$/. Since the candidate is impossible in Tagalog, we seek an analysis in which it gets a probability of near-zero. This will only be the case if UNIF-/maŋ-$_{other}$/ is given a very substantial weight, with NASSUB and the remaining UNIFORMITY constraints comparably boosted.

In light of such cases, we recommend careful attention in cases of zero weights: if the system is considered more broadly, does it turn out that the zero-weighted constraint has other uses? These often can be demonstrated by adding candidates or inputs to the analysis.

This said, we add that sometimes, a constraint is fit with a weight of zero simply because that is the actual best-fit weight for the training data. A schematic case of this is shown in (61).

(61) *A case in which a weight must be zero to fit the data*

			$\mathbb{C}_1$ **1.099**	$\mathbb{C}_2$ **0**	$\mathcal{H}$	eH	Z	p
Input$_1$	*Cand$_1$*	0.75			0	1	1.33	0.75
	Cand$_2$	0.25	1		1.10	0.33		0.25
Input$_2$	*Cand$_1$*	0.75			0	1	1.33	0.75
	Cand$_2$	0.25	1	1	1.10	0.33		0.25

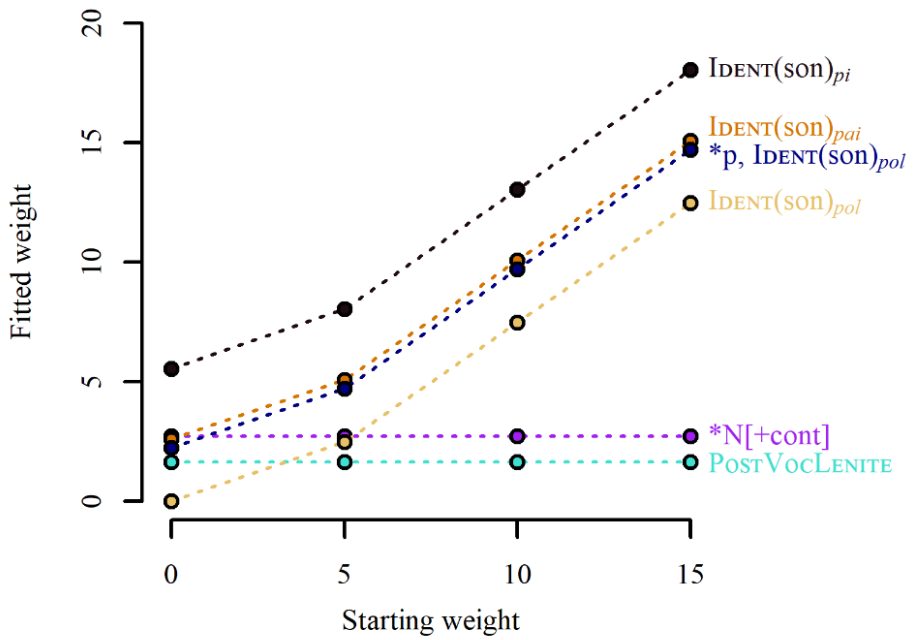
Here, any non-zero weight assigned to $\mathbb{C}_2$would result, incorrectly, in the assignment of different output probabilities to *Input$_1$* and *Input$_2$*.

How can the analyst tell whether a constraint's weight must be zero or not? We suggest two methods. First, in Solver, one can fix the weight of a constraint in advance at a non-zero value (see Figure 29 for how), run the Solver to fit the remaining constraints, and see if the same model fit can be achieved. For (61), if we fix the weight of $\mathbb{C}_2$ at 1 and solve, the log likelihood worsens from −1.12 to −1.17. The predicted probabilities are a worse match than those shown in (61) as well, with 0.66 predicted for *Cand$_1$* of *Input$_1$*, but 0.84 predicted for *Cand$_1$* of *Input$_2$*. In contrast, for the Tagalog case, setting the weight of UNIF-/maŋ-$_{other}$/ to 1 and re-solving results in exactly the same model fit as in the tableau in (60).

[22] The reader may object that [ãmigaj] could be ruled out independently by a ban on nasalized vowels, which are not part of the Tagalog phoneme inventory. But the very same point could be made with [amigaj], in which non-nasal [a] is also the result of coalescence.

A second method is to run the Solver using a few different starting weights. This can serve as a useful diagnostic not just of zeros but of the model structure overall.[23] For instance, we tried fitting weights to our K5 model (§2.7) starting all constraints out with a weight of 0, 5, 10, or 15. All of the resulting models were equally accurate (log likelihood = −267.54), but the constraint weights often varied depending on their starting point. For the perturber constraints *N[+cont], and POSTVOCALICLENITE, the weights did *not* vary, as shown in Figure 19. Evidently, these weights must have their original fitted values to match the data. But the remaining constraints responded to varying starting points by varying in parallel; for them is it is the *differences* among weights (specifically, pairwise between *p and each IDENT constraint) that matters.

Figure 19: How the weights for K5 emerge with different starting points



We find it helpful to examine the "elsewhere" inputs in our K5 tableau ((31) above, p. 34) in light of this patterning — these show the pairwise violations that imply a constant weight difference.

To conclude this section, we return briefly to our earlier discussion (§4.3) of the differences between a MaxEnt grammar and a statistical analysis; we here consider zero weights from this perspective. In both (59) (schematic case) and (60) (Tagalog), one of the parameters of the model is — in statistical terms — redundant; you can predict the output patterns just as well leaving one parameter out. If we attempted to fit these models using statistical software such as R, we will get an error; in R the program provides an error message stating that the model is not 'identifiable', meaning that there is no unique solution — just as described above. Thus, it is yet another difference between MaxEnt OT and statistical testing is that it is only in the former that redundant constraints are tolerated. Of course, even though there are good scientific reasons to

[23] A note of caution: this procedure will not work if one is using a prior, as discussed in §4.6.

employ redundant constraints, it pays for analysts always to be aware of such redundancies when they interpret their weights.

4.5 The math of weight-fitting

In Chapter 2, we introduced the use of Excel Solver for fitting weights. Here we go into the math in a greater depth, covering first the essential notion of likelihood, then covering the results that make trustable weight-setting possible within MaxEnt.

4.5.1 More on likelihood

As covered in Chapter 2, we use log likelihood as a metric for the accuracy of a MaxEnt analysis, and as the basis for finding the best-fit weights for the data. We demonstrated how to calculate and use log likelihood in §2.6.4, but we now go into more detail about why log likelihood is a sensible metric of accuracy.

For intuitive purposes, it is best to start not with log likelihood, but with likelihood per se. As noted above, the likelihood of the data is the probability that our data would be observed if our model is correct. The idea is that if we are comparing the behavior of several models on a single dataset, the best one is the one that makes the actual observed data the most likely.

Likelihood is calculated by building up from the probabilities that the model assigns to single data points. For a simple case, we return again to K3, from §2.4. To recall the facts, in K3 /p/ lenition is blocked after nasals, inapplicable in stems, and applies 73% of the time in all other cases; the data are repeated below in (62). Earlier, we developed a three-constraint MaxEnt grammar to cover these data, whose constraints and weights are relisted in (63).

(62) *Hypothetical data for K3*

UR	SR	count
/N-p/	N-p	212
	N-w	0
/C-p/	C-p	300
	C-w	110
/C-$p_{content}$/	C-p	720
	C-w	0

(63) *MaxEnt grammar for K3*

Constraint	*Weight*
IDENT(son)$_{content}$	20
*N[+cont]	20
*p	3
IDENT(son)	4[24]

In order to understand likelihood better, let us begin with a single data point, an instance of the observation [mʊntəg **w**oɮən] from underlying /mʊntəg **p**oɮ-ən/ (see (18)). The grammar we developed for K3 assigns a probability of 0.27 to this outcome. According to this grammar, then, the likelihood of our observation is exactly 0.27.

Suppose we expand our data set to two forms, of which the second is an instance of [mʊntəg **p**oɮən], without lenition, from the same input. Our grammar assigns a probability of 0.73 to [mʊntəg **p**oɮən] and of 0.27 to [mʊntəg **w**oɮən]. Here, the likelihood of the data for this expanded data set is 0.73 × 0.27 = 0.1971. This is because the probability of two independent events is their product (e.g., the probability of two heads in a double coin flip is 0.5 × 0.5 = 0.25). Observe that there is another grammar that would better match this tiny dataset: it would assign 0.5 probability to both [mʊntəg **p**oɮən] and [mʊntəg **w**oɮən]. Its likelihood would be higher, i.e. 0.25.

Scaling up further, suppose now a still larger subset of the original data, as in (64):

(64) *Pseudo-data for K3 (likelihood illustration)*

/mʊntəg **p**oɮən/	→	[mʊntəg **p**oɮən]	15 tokens
/mʊntəg **p**oɮən/	→	[mʊntəg **w**oɮən]	5 tokens

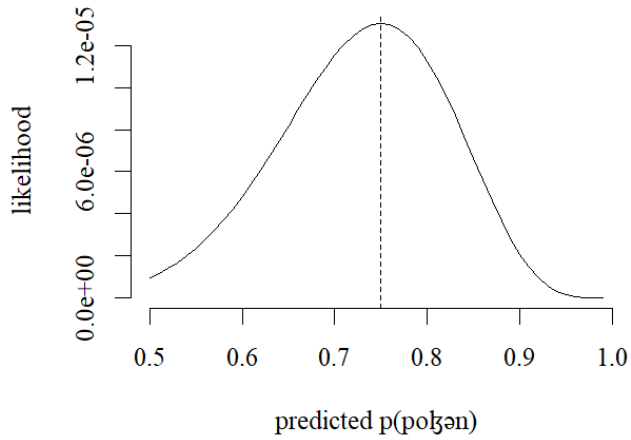
Probability assigned to [mʊntəg **p**oɮən] by grammar (63):	0.73
Probability assigned to [mʊntəg **w**oɮən] by grammar (63):	0.27

Again, we multiply out the probabilities for all 20 data, using exponentiation to save space. This turns out to be $0.73^{15} \times 0.27^{5} = 1.278 \times 10^{-5}$.

Once again, there is an intuitively better grammar available: we find weights such that P([mʊntəg **p**oɮən]) is exactly 0.75 (matching 15/20) and P([mʊntəg **w**oɮən]) is exactly 0.25 (matching 5/20). The likelihood for this grammar turns out to be 1.305×10^{-5} — a bit higher, reflecting our intuitions. More generally, we can calculate the likelihood of the data for a whole range of values of P([mʊntəg **p**oɮən]), showing how the maximum is indeed at 0.75. This is seen in Figure 20:

[24] Recall that our weights for K3 were fit by hand; this is why they are whole numbers.

Figure 20: How the likelihood of data (64) depends on P([mʊntəg **p**oɫʒən]*)*



The graph offers some insight into the problem that likelihood addresses. If we were to propose a very high probability for [mʊntəg **p**oɫʒən] (right side of graph), we would make this, the most common outcome, more likely — the likelihood for just the [mʊntəg **p**oɫʒən] data points would be close to 1, which is good. But the better we do at predicting the [mʊntəg **p**oɫʒən] data points, the worse we do at predicting the [mʊntəg **w**oɫʒən] data points. The likelihood criterion allows us to find the best balance across all of our data, which matches the overall 0.75/0.25 observed distribution.

The general procedure for calculating likelihood for any size of dataset is given in (65).

(65) *Procedure for calculating likelihood*

a. Find the probability that each observed datum is assigned by the analysis. For data types (identical violation profile) that occur multiple times in the training data, count the *n* tokens of this type as *n* separate instances.
b. Multiply these probabilities together. This product is the likelihood.

Expressed as a formula, likelihood is defined as in (66).

(66) *Likelihood formula*

$$\text{Likelihood} = \prod p_i^{n_i}$$

where

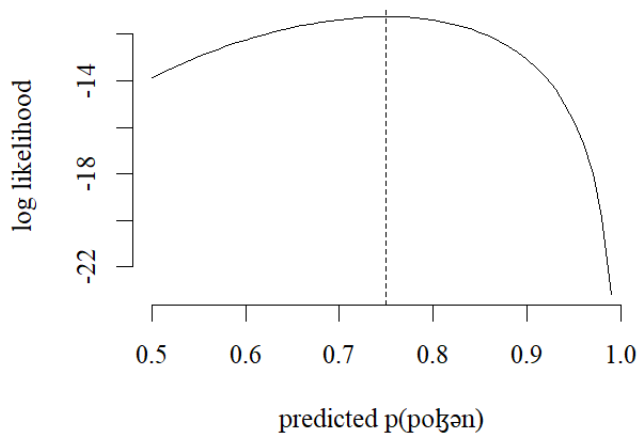
n_i is the number of tokens of output *i*
p_i is the probability the model assigns to output *i*
Π denotes multiplication

Having covered likelihood, it is straightforward to extend the concept to log likelihood, which is what we actually use for purposes of weight-fitting. When we scale up to larger cases we will find that, the actual likelihood — although fully meaningful as a measure of model fit — becomes extremely small. For example, the likelihood of the K4 data in (20) under our final

analysis is 5.31×10^{87}. As we move beyond K4 (which is still just a toy example), to cases of realistic size, the software we use will no longer be able to represent the likelihood and will round it off to the (useless) value of zero. We can overcome this problem by introducing logarithms: the *log likelihood* of a dataset under an analysis is the log of the likelihood.

For purposes of setting the weights, log likelihood will serve just as well as likelihood: because log likelihood increases *monotonically* with likelihood. Hence, whatever weights maximize log likelihood will also maximize likelihood. To show this, the graph of Figure 20 is replotted below in log-likelihood terms, showing that we get the same maximum point.

Figure 21: Log likelihood yields the same maximum point as likelihood



Thus, using logarithms makes the numbers more tractable, but otherwise works very similarly to likelihood and yields the same answer. In (67) we give the math for calculating log likelihood.

(67) *Deriving equivalent calculations for log likelihood*

$\text{Likelihood} = \prod p_i^{n_i}$	from (65)
$\log(\text{Likelihood}) = \sum \log\left(p_i^{n_i}\right)$	because $\log(x \times y) = \log(x) + \log(y)$
$\log(\text{Likelihood}) = \sum n_i \log\left(p_i\right)$	because $\log(x^y) = y \times \log(x)$

As can be seen, whereas in calculating likelihood we dealt with *n* instances of a datum by taking its probability to the *n*th power, with log likelihood we multiply by *n*. Whereas for likelihood we dealt with multiple types of data by multiplying, for log likelihood we sum them. These are the calculations we have already been doing on the spreadsheets given in Chapters 2 and 3. For example, in (24) on p. 29 above, n_i corresponds to column D, $log(p_i)$ to column N, and the computation itself is carried out by the `SUMPRODUCT()` function in cell `O2`.

Excursus: Interpreting log likelihood values

The log likelihood of a model is not directly interpretable as a number; it only is useful as a basis of comparison. Indeed, log likelihood values will be very different across simulations because it responds directly to the size of the data set. But even though log likelihood is

arbitrary, it is possible to calculate a couple of anchor points that can give us a rough idea of how well an analysis is doing.

First, if we set all constraint weights to zero, we can get a baseline likelihood, essentially the equivalent of saying there is no grammar at all: the candidate probabilities for each input come out the same. We will call this the **zero-model**. Obviously, zero models are uninformative and receive very low log likelihood. We can use them as a lower anchoring point.

At the other end of the scale, we can construct an equally absurd grammar that obtains the highest possible log likelihood for a given data set. For instance, one could construct a model that includes a constraint of the form INPUT X MUST HAVE OUTPUT Y for all combinations of X and Y. Such a model is often called a **saturated model**.

Note that, although a saturated model is likely to have a high log likelihood, the value will *not* be as high as zero. A zero log likelihood will arise only when there is no variation in the data, and the grammar matches the data perfectly. In other words, there is a difference between perfectly matching every single data point and perfectly matching a probability distribution; it is the latter goal that we are generally aiming at in this book.

The saturated model does not actually need to be constructed; rather one can substitute "empirical probabilities" for predicted probabilities in the log likelihood equation. By empirical probability, denoted E_{ij}, we mean the fraction of the observed cases with input *i* in which the output is candidate *j*. Using this concept, we can calculate the log likelihood of a saturated model as in (68).

(68) *Formula for log likelihood of a saturated model*

$$\sum n_{ij} \log(E_{ij})$$

where

Σ denotes summation over all candidates for all inputs
n_{ij} is the number of times candidate j for input *i* is observed
E_{ij} is defined as above.

A sample spreadsheet that performs these calculations for our K5 analysis is given in the Supplementary Materials as **K5_zero_and_saturated.xlsx**.

Below, we give for three models discussed in this book, the log likelihoods of the zero model, the model we proposed, and the saturated model.

(69) *The log likelihood of three models, as bounded by the random and saturated models*

	Zero model	*Our model*	*Saturated model*
K5 (§2.7)	−482.4	−267.5	−261.5
Tagalog (§3.1)	−600.3	−248.6	−221.2

Turkish (§6.2)	−31199.9	−25317.3	−24410.9

We see here that in the absence of comparison, log likelihood is meaningless; the Turkish simulation just happens to have far more data points than K5 or Tagalog.

A final note: in calculating likelihood we need to consider the possibility that the calculation might crash due to the presence of zeros. First if the *observed* frequency is 0, we needn't worry: the log of the predicted probability will be multiplied by zero, adding zero to the total log likelihood. On the other hand, if any *predicted* probability were truly zero we wouldn't be able to calculate likelihood — the log of zero is undefined, so the total log likelihood would also be undefined. It doesn't help to go back to plain (non-logged) likelihood, because we would wind up multiplying all other probabilities by zero, yielding a likelihood of zero no matter what the models' predictions were for other candidates. Fortunately for us, MaxEnt never generates true zero as a predicted probability, only probabilities very close to zero, so this issue will generally not arise. It is, however, a tricky issue for other theories of probabilistic phonology (see §9.5) which do generate actual zeros.[25]

4.5.2 Convexity and finding the maximum likelihood weights

Having shown that maximizing log likelihood is a sensible way to find an accurate grammar, we turn to the problem of finding a set of weights that achieve this maximum. To facilitate understanding how these weights are found, it is helpful to consider the problem in geometric terms. Imagine a simple two-constraint grammar, along with a dataset it is intended to model.

(70) *Toy grammar for illustrating weight computation*

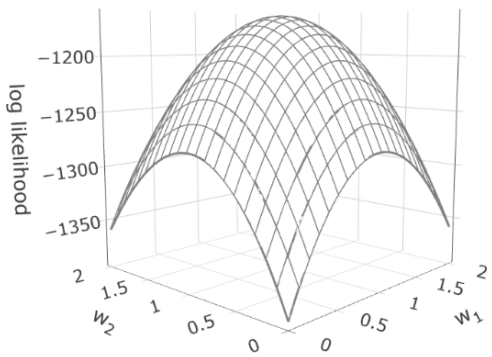
		obs.	$\mathbb{C}_1$ $\mathbf{w_1}$=1	$\mathbb{C}_2$ $\mathbf{w_2}$=1	$\mathcal{H}$	eH	Z	p	ln p	LL
Input$_1$	*Cand$_1$*	731			0	1	1.368	0.731	−0.313	−1164.5
	Cand$_2$	269	1		1	0.368		0.269	−1.313	
Input$_2$	*Cand$_1$*	731			0	1	1.368	0.731	−0.313	
	Cand$_2$	269		1	1	0.368		0.269	−1.313	

As can be seen, Input$_1$ and Input$_2$ behave identically with $\mathbb{C}_1$ and $\mathbb{C}_2$ respectively.

To visualize the weight computation (as carried out, for instance, by the Solver) we can make a three-dimensional graph, with the two constraint weights on the *x* and y axes, and the log likelihood corresponding to every possible combination of weights appearing on the vertical *z* axis. A graph of this type is shown in Figure 22.

[25] One way of dealing with this is simple to convert all zero model probabilities to some very small value; Zuraw and Hayes (2017) use 0.001.

Figure 22: 3D graph relating log likelihood to the weights of a two-constraint grammar



Here, the log-likelihood surface forms a dome, whose highest point (altitude −1164.5) occurs where w_1 and w_2 have their optimum values, in this case $w_1 = 1$, $w_2 = 1$. With visualizations like this, one can construe the problem of finding the best weights as a hill-climbing problem: keep going uphill until you can't any longer. Of course, when the number of constraints is higher than two, then the hill becomes many-dimensional and impossible to visualize, but the computation is essentially the same.

The efficient ascent of hills like in Fig. 22 is a standard problem in computer science, and many algorithms exist, including, for instance, the one in the Excel solver (Generalized Reduced Gradient Method; Lasdon et al., 1974). Generally, the algorithms proceed iteratively, leaping up the hill in stages, each calculated by a formula that defines the algorithm (more on this below). In larger problems you can see the stages proceeding one by one at the bottom of your screen when operating the Excel Solver.

For ordinary linguistic analysis, we can sensibly just pick some widely-used algorithm; it should not be a matter of linguistic theory to invent new hill-climbing algorithms. However, an encouraging fact about hill climbing in MaxEnt is that the search space is very well-behaved: there is never more than one peak, so that for any algorithm, it suffices simply to keep going uphill; you will never get stuck in a so-called local maximum. Another way this is often put is to say that the log-likelihood function for MaxEnt is CONVEX. A proof of convexity for MaxEnt is given in Della Pietra et al. (1997), and explained in textbook form by Jurafsky and Martin (2026: §4.6) . As we will see in §9.5, the convexity of the MaxEnt search space has been taken as a possible advantage of MaxEnt over similar theories.

Excursus: Slope as a means of effective weight-fitting

As noted in the main text, search algorithms for MaxEnt generally climb the hill of log likelihood in a series of uphill leaps (see Hayes and Wilson 2008: Figure 2 for a visual example). Just what direction to leap and how far is part of the art of algorithm design. To make the leaps effective, it is useful to know the *slope* of the hill at any one point, by which we mean both the uphill direction and the steepness. Calculus gives a specific meaning for slope: it is the vector of partial derivatives of the objective function against the constraint weights.

As it turns out, the slope of the MaxEnt log likelihood at any point P can be expressed by a very simple formula: it is the number of Observed violations of the constraint in the data, minus the number of Expected violations (O − E). The latter can be calculated from the predicted probabilities of all candidates using the set of weights defined at P.

Many search algorithms, including the one used in the Excel Solver, use the slope to calculate what would be an effective size and direction for each uphill leap. Note that at the top of the hill, the slope is zero, and so O = E for all constraints. In other words, every constraint is violated in the aggregate exactly as often as the optimized grammar says it should be.

For more details on the use of slope in weight-fitting, including a proof of the "O – E" theorem, see Jurafsky and Martin (2026: Chapter 5).

4.5.3 Search spaces with no unique maximum

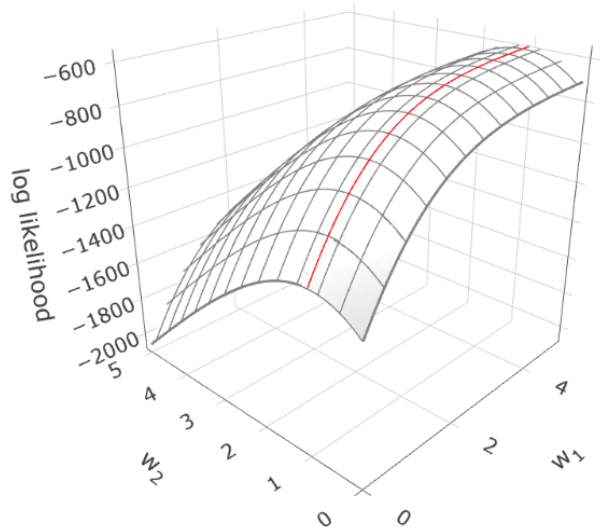
The discussion above is somewhat idealized: while the MaxEnt search space is convex in the mathematical sense, it is not always a pretty dome; and the exceptions relate to two scenarios we have already discussed. Consider first the case where one of the best-fit weights has an infinite value (§2.9, §4.4.1). This can be intuitively understood if we draw the hill for this scenario, based on the tableau in (71). This tableau is just like (70), except that Input_2 yields exclusively $Cand_1$ as its output, necessitating an idealized weight of infinity for $\mathbb{C}_2$.

(71) *Toy grammar to illustrate weight computation with infinite ideal weights*

		obs.	$\mathbb{C}_1$ $\mathbf{w_1}$=1	$\mathbb{C}_2$ $\mathbf{w_2}$=∞	$\mathcal{H}$	eH	Z	P	ln p	LL
Input_1	$Cand_1$	731			0	1	1.368	0.731	−0.313	−582.262
	$Cand_2$	269	1		1	0.368		0.269	−1.313	
Input_2	$Cand_1$	1000			0	1	1	1	0	
	$Cand_2$	0		1	∞	~0		~0	−∞	

The corresponding search-space hill is given in Figure 23.

Figure 23: Hill of search for a two-constraint grammar, one weight infinite



With regard to weight $\mathbf{w_1}$, the hill is the same as in Figure 22, since for any value of w_2, the highest log likelihood will fall where $\mathbf{w_1} = 1$. But for weight $\mathbf{w_2}$, the hill forms a ridge, extending off to infinity, with ever-decreasing slope along the line of the ridge, where log likelihood asymptotes to its maximum value of −1164.5 at (1, ∞). The search space is still convex in the mathematical sense, but it never attains a single best-fit value.

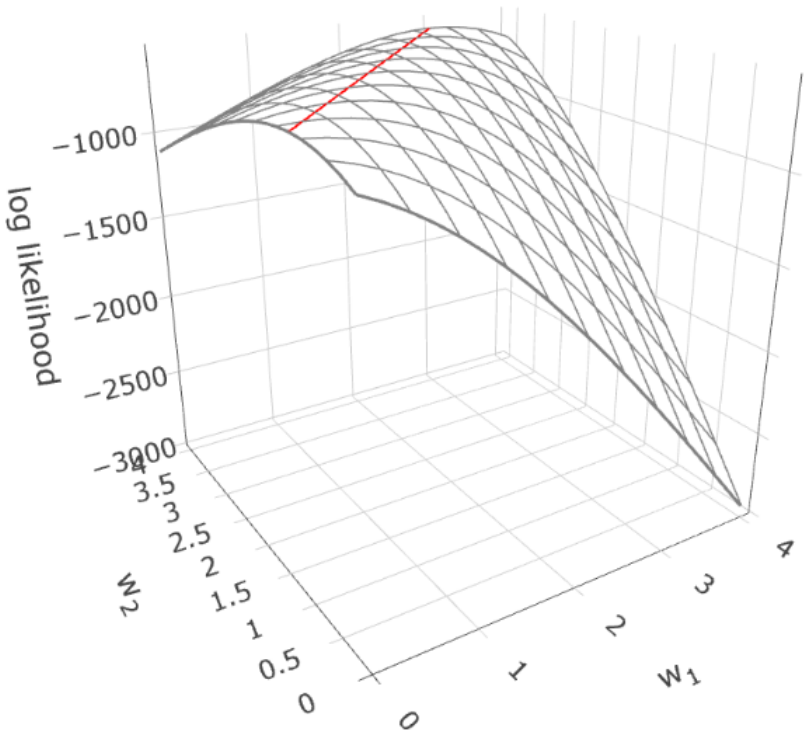
A second case not resembling a simple dome arises from a point we made earlier in §4.4: often there is not one unique setting of the weights that achieves maximum likelihood, because what actually matters is the sum or difference of the two weights. For the case of constant difference, a representative tableau is given in (72).

(72) *A grammar where only the difference between weights matters*

		obs.	$\mathbb{C}_1$ $\mathbf{w_1} = 1.5$	$\mathbb{C}_2$ $\mathbf{w_2} = 0.5$	$\mathcal{H}$	eH	Z	p	ln p	LL
Input$_1$	*Cand$_1$*	731		1	0.5	0.606	0.830	0.731	−0.313	−582.2
	Cand$_2$	269	1		1.5	0.223		0.269	−1.313	

Tableau (72) is similar to (59) above; a perfect fit will be achieved as long as weight $\mathbf{w_1}$ is precisely 1 higher than $\mathbf{w_2}$. When we visualize the search space (Figure 24), we see a flat ridge, marked in red, corresponding to the set of maxima where $\mathbf{w_2} = \mathbf{w_1} + 1$.

Figure 24: Hill of search for (72)



In addition to our values of (1.5, 0.5), the infinite ridge also includes the points (1, 0), (2, 1), (3, 2), (4, 3), and so on. Any position along the ridge yields the same, optimized, log likelihood. An adequate search algorithm can be trusted to climb up to the ridge, but where along the ridge it ends up will depend on the starting point and particularities of the algorithm.

4.6 An introduction to priors

In MaxEnt, weight fitting is often carried out with the use of a **prior**. In Chapter 7, we discuss the use of priors as a theoretical tool; here we simply introduce the concept, focusing on the problem raised by infinite weights. Let us return to example (71), where the best-fit weight of the second constraint is infinite. We repeat that tableau here, in (73).

(73) *Tableau (71) (repeated)*

		obs.	$\mathbb{C}_1$ $\mathbf{w_1}$=1	$\mathbb{C}_2$ $\mathbf{w_2}$=∞	$\mathcal{H}$	eH	Z	P	ln p	LL
Input$_1$	*Cand$_1$*	731			0	1	1.368	0.731	−0.313	−582.262
	Cand$_2$	269	1		1	0.368		0.269	−1.313	
Input$_2$	*Cand$_1$*	1000			0	1	1	1	0	
	Cand$_2$	0		1	∞	~0		~0	−∞	

Recall that the goal here is to assign to constraint $\mathbb{C}_2$ a weight that is high enough to approximate the 1-0 frequency distribution, without crashing our weight-fitting software. Whatever procedure we follow, it should also be able to weight $\mathbb{C}_1$ in a way that matches the frequencies of Input_1 correctly.

In §2.10.1 we suggested the practice, when using the Excel Solver, of placing an upward limit of 50 on constraint weights, with precisely problems like this in mind. This suffices as a rough-and-ready strategy, but it is not ideal, since the limit of 50 is arbitrary. In some cases, we may indeed need weights to be higher than 50, while in other cases a much lower weight could achieve essentially the same results. What we really want is a way to tell the Solver, "make these weights as high as you need to get close to correct predictions for the data, but no higher".

A **prior** is a more reliable method for limiting constraint weights which works in the general case, though it takes more effort to implement. To define a prior, let us first introduce a simple notion: the **objective function**, which is any computed value that we maximize (or for some functions, minimize) for purposes of fitting the weights of a model, such as a MaxEnt model. So far, the objective function we have been using has been log likelihood. This objective function values nothing but accuracy. Here we will introduce a slightly different objective function that sacrifices a bit of model accuracy to keep constraint weights at a reasonable level.

We start with a very simple prior: we sum the constraint weights, as in (74). This sum is then subtracted from the log likelihood to form the objective function.

(74) *An objective function with a simple prior (sum of all weights)*

a. Prior = $\sum_{i \in constraints} w_i$

b. Objective function = log likelihood − prior

The inclusion of the prior disfavors high constraint weights. This is because it is subtracted from log likelihood to form the objective function, which is maximized. Implementing it with the Solver, we obtain the weights $\mathbf{w_1} = 0.995$, $\mathbf{w_2} = 6.907$. These are lower than what we fitted without the use of a prior, which were $\mathbf{w_1} = 1.000$, $\mathbf{w_2} = 18.094$. Notice that it is $\mathbf{w_2}$, whose ideal value is infinite, that responds more strongly to the prior.

Source: C:\Dropbox\AR\MaxEntBook\PrepareSims\Chapter4\PrimitivePrior.xlsx.

The prior defined in (74) can be considered as implementing an *a priori* preference that weights be close to zero. The prior defines the point at which an increase in accuracy of model fit is assumed to not be "worth" a continued increase in constraint weights.

4.6.1 The Gaussian prior

A more commonly used prior is the **Gaussian** or **L2** prior, given in (75) below.

(75) *Calculating a Gaussian prior*

a. Square every constraint weight.
b. Sum the values obtained in (a) for all constraints.
c. Divide this sum by a user-chosen value. For reasons to be given, this value is expressed as $\mathbf{2\sigma^2}$.

Expressed as a formula, this is:

$$\sum_i \frac{w_i^2}{2\sigma^2}$$

As before, the prior is subtracted from the log likelihood to obtain the objective function.

(76) *Objective function incorporating a Gaussian prior*

$$\text{Objective} = \text{log likelihood} - \sum_i \frac{w_i^2}{2\sigma^2}$$

Excursus: Terminology and detailed formation of the Gaussian prior

It is natural to wonder why the prior in (76) is called "Gaussian"; and also why the formula includes a non-obvious scaling method (why $2\sigma^2$, when just σ should suffice?). The two questions turn out to have the same answer. The MaxEnt objective function is normally defined in the domain of log likelihood. This is done only for computational convenience, and we can just as well switch it back to simple likelihood (§4.5); the function *exp*() switches it back. Doing this, we find that the contribution to log likelihood from the Gaussian prior, for any one constraint i, is $\exp(w_i^2/2\sigma^2)$. This expression is, in fact, the standard formula for a Gaussian distribution (the familiar "bell curve") of a random variable w_i, with mean zero and standard deviation σ. This is why the Gaussian prior is so-called, and it also gives a rationale for dividing by $2\sigma^2$: $\exp(w_i^2/2\sigma^2)$ establishes a *prior probability distribution* for the weight of constraint i, which has its maximum at zero, and a standard deviation of σ.

In Chapter 7, we will find it useful to work with more complex priors that give each constraint its own preferred weight, designated as μ. For these priors, the probability distribution becomes as in (77):

(77) *Gaussian prior for assigning preferred weights to individual constraints*

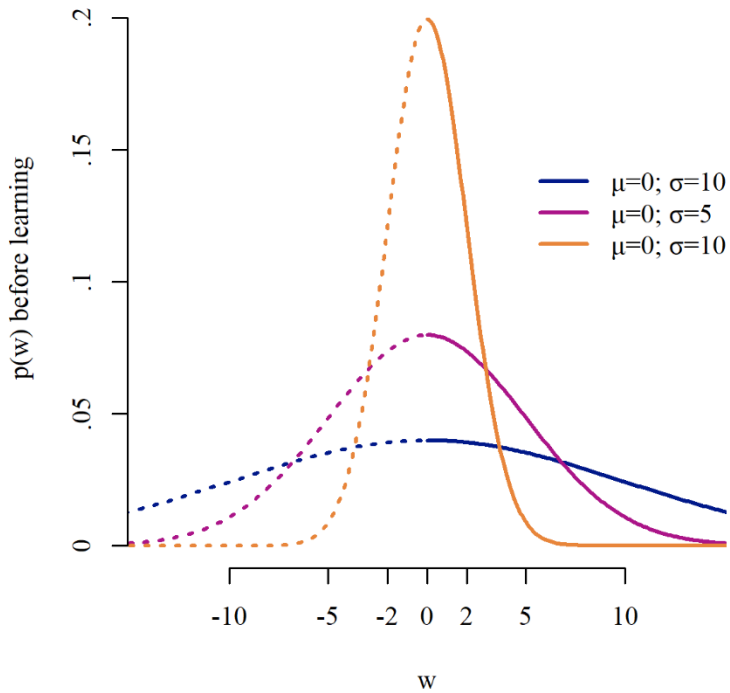
$$\sum_i \frac{(w_i - \mu_i)^2}{2\sigma^2}$$

where μ_i is the preferred weight of the ith constraint

The parameter μ expresses the mean of the Gaussian curve, and σ expresses its standard deviation. Three different Gaussian priors are illustrated for a one-constraint grammar in Figure 25; these priors all prefer the weight of the constraint to fall at or near μ, which is always 0, with

strictness governed by σ, which is assigned the values 2, 5, and 10. As before, dotted lines are used for negative weights.

Figure 25: Three Gaussian priors



Note that for smaller values of σ, the main body of the Gaussian becomes narrower, expressing a stronger a priori preference for a weight close to 0.

We turn now to the implementation of a Gaussian prior in Excel. The spreadsheet **AvoidingInfiniteWeights.xlsx** is illustrated in (78). The formulas (rather than values) of the crucial cells are shown in `Courier`. The weights given in cells D2-E2 are in fact the weights that are delivered by the Solver when we set the weights under this arrangement.

(78) *A spreadsheet illustrating the use of a Gaussian prior*

	A	B	C	D	E	F	G	H	I	J	K
1				C1	C2						
2			Weight:	**0.995**	**5.25**	H	eH	Z	p	ln p	Log Lik.
3			Squared:	=D2^2	=E2^2						
4	Input$_1$	*Cand$_1$*	731			0	1	1.370	0.730	-0.3147	**-587.5**
5		*Cand$_2$*	269		1	0.995	0.370	1.370	0.270	-1.3094	
6	Input$_2$	*Cand$_1$*	1000			0	1	1.005	0.995	-0.0053	
7		*Cand$_2$*	0		1	5.245	0.005	1.005	0.005	-5.2504	
8	=SUM(D3:E3)	(75a,b): Sum the squares of the weights.									
9	1	We chose this value for sigma.									

10	=2*A9^2	Square and double.
11	=A8/A10	(75c): Divide A8 by A10, yielding the Gaussian prior.
12	=K4	Log likelihood (repeated)
13	=A12-A11	Subtract A11 from A12, yielding the objective function (−601.8)

As before, fitting with a prior results in lower weights. Here the best-fit weights are $\mathbf{w_1}$ = 0.995, $\mathbf{w_2}$ = 5.25; without a prior the best-fit weights are $\mathbf{w_1}$ = 1.000, $\mathbf{w_2}$ = infinite.[26] The lowering effect depends in general on the value of σ, with higher σ lowering the weights less.

In (78), we set our prior with a fairly low value of σ (i.e., 1) in order to strongly favor low weights. With this low σ, we can see that priors do sacrifice some model accuracy; in this case $Cand_1$ for Input1 was assigned a probability of 0.730 instead of 0.731; and $Cand_2$ for Input2 was assigned 0.995 instead of 1.000. Had we used a larger value of σ, these discrepancies would have been smaller. At the other end of the scale, a tiny σ value like 0.0000001 would force the weights almost to zero, so the candidates would come out nearly equiprobable − a null grammar. Even a strong prior can of course be overcome by large amounts of data pointing to a non-zero weight for a given constraint; thus an appropriate value of σ depends on the size of the dataset.

The choice of σ also depends on the use that is being made of the prior. In the present context, all we are trying to do is avoid infinite weights, and for this purpose even a very weak prior (= very high σ, like 100,000) will suffice. In Chapter 7, we will encounter a different use of priors in which the choice of σ is a theoretically informed decision.

For why we might favor a Gaussian prior vs. the simpler prior in (74), see the following excursus box.

Excursus: The L1 prior

The simple prior we discussed initially in (74) is an approximation of the so-called "L1" prior, in which the weights are not squared; in fact the Gaussian prior is sometimes called the "L2" prior precisely because weights are squared. The standard version of the L1 prior includes an absolute value sign in order to penalize negative weights that deviate far from zero. It also includes a multiplicative constant α, so the prior is can be made stronger by adopting a larger value for α.

(79) *The L1 Prior*

$$\alpha \times \sum_i |w_i - \mu_i|$$

[26] Infinity is indeed the best-fit weight for $\mathbf{w_2}$ without a prior, but the Solver actually comes up with 18.09. There is evidently something in its algorithm, not diagnosed by us, that sometimes prevents it from running lemming-like towards a fatal unbounded-weight crash. That the Solver algorithm does sometimes leap over the cliff was shown above under (36), where the use of a prior (or weight limit) is truly essential for weight-fitting to succeed.

When would we want to use an L1 vs. an L2 prior? The squaring used in the L2 prior has the effect that weight tends to be shared across constraints that overlap in their effects; L1 in contrast tends to concentrate weight on as few constraints as possible. The sharing tendency of the L2 prior is used to interesting effect by Martin (2011) to explain the so-called "leakage" effect, where (for instance) constraints that are in principle applicable only word-internally also end up having modest phrasal effects.

4.6.2 Priors and replicability

Adopting a modest prior helps the *replicability* of an analysis, because it generally forces a unique weighting, circumventing the difficulties shown in §4.6 above. To illustrate this, we re-ran our example (71) using different software, the `maxent-ot` R package of Mayer et al. (2024). The results, compared with Excel, are shown in (80).

(80) *Increasing replicability by using a prior*

	Weight of $\mathbb{C}_1$	*Weight of* $\mathbb{C}_2$	*p(In1, Cand1)*	*p(In2, Cand1)*
No prior:				
Excel with Solver	0.9996978	20.302	0.730992000	1.00000000
`Maxent.ot`	0.9997019	18.094	0.730999970	0.99999999
Difference	**0.0000041**	**2.208**	**0.000007970**	**0.00000001**
Gaussian prior, σ = 1				
Excel with Solver	0.9946501	5.245190	0.730005430	0.99475483
`Maxent.ot`	0.9946498	5.245181	0.730005373	0.99475479
Difference	**0.0000003**	**0.000009**	**0.000000057**	**0.00000005**

Since under a prior, the two software packages are optimizing the exact same problem, their answers come out extremely similar, far more similar than without the prior (compare boldface rows). Without the prior, it appears Maxent.ot, in its default settings, tolerates a bit more imperfection in model fit and so stops with w_{C1} at 18.094.[27]

We find the use of prior here to be comforting in a different sense: the solution you get is mathematically well-defined. Without a prior, the solution you get is unprincipled, in the sense that (in its details) it depends on arbitrary aspects of the weight-setting algorithm.

[27] Thanks to Connor Mayer for providing this explanation.

Excursus: Gaussian priors ensure dome-shaped search spaces

When we employ a Gaussian prior, the search spaces corresponding to Figures 23 and 24 above lose their anomalous character and appear instead as dome-shaped. The reason is that the surface corresponding to the Gaussian prior is itself a dome, specifically an inverted paraboloid. When this surface is added to the ridgeline shape of Figure 24, for example, it "deflattens" the objective function, providing a unique maximum. The slope of the paraboloid along either axis turns ever more sharply downward as we move outward from the point of origin. This means that in cases like Figure 23, no matter how weak the prior (i.e. how high σ is), it will eventually create a downward slope for the combined objective function, again yielding a unique maximum.

All this is in effect a geometric restatement of the point just made: Gaussian priors help make the choice of weights less arbitrary in scenarios when many weightings fit equally well or almost equally well.

4.7 History and etymology of "MaxEnt"

As noted in §2.12.3, the first linguists to use the math of MaxEnt for theoretical purposes were Rousseau and Sankoff (1978), where the math was used to implement a probabilistic version of the phonological theory of Chomsky and Halle (1968). In this "variable rules" approach, each rule comes with a little MaxEnt grammar with just two candidates, "Apply rule" and "Don't apply rule". The framework proved influential and inspired a large literature, notably work that employed Sankoff's Varbrul software.

A version of MaxEnt more recognizable from the perspective of this text appeared in the early work of Paul Smolensky (Smolensky 1986). Interestingly, Smolensky at the time had no particular interest in language, but was using "knowledge atoms" (for us, constraints) as ingredients of a general theory of cognition. His article includes recognizable versions of weights, Harmony (Smolensky's own term), eH, and Z. It is widely cited in cognitive science.

Maximum Entropy Optimality Theory, the full name of the framework we are discussing, arose in the early years of this century when computer scientists noticed that mathematical tools that they were already using could be bolted onto OT to form an effective stochastic theory. Among these scholars were Johnson (2002), Eisner (2000), Manning (2003), and most influentially, Goldwater and Johnson (2003), who gave the framework its name and applied it effectively to previously-studied examples.

In the Introduction to this book, we noted that we are discontented with the term "MaxEnt" but use it in obedience to tradition. In the present context, we observe that our "MaxEnt" is actually the combination of two things: a calculation system for deriving candidate probabilities from weights and violations, and a system for setting the best-fit weights. Only the second of these has any actual connection with maximizing entropy, and indeed this system is compatible with other stochastic frameworks as well (see §9.5 below). In reading discussions of MaxEnt in the literature, it is helpful to identify which of these two aspects of the approach the author is

addressing. For detailed discussion of how the concept of maximum entropy (as used in information theory) is applicable to MaxEnt, see Jäger (2007).

Chapter 5: Research technique

MaxEnt is a tool for research, and in this chapter we cover three aspects of linguistics research that bear on MaxEnt: first, the use of graphing and statistical methods to assess a MaxEnt analysis (§5.1-§5.3); second, the choice and deployment of constraints in the development of an analysis (§5.4), and lastly a brief survey of the methods used to gather data that can be analyzed in MaxEnt (§5.5).

5.1 Assessment of analysis in MaxEnt

In traditional generative linguistics (e.g. starting with Chomsky, 1957; 1965; Chomsky and Halle, 1968), analyses are evaluated based on whether they *work* (i.e., cover the relevant set of empirical data) or *don't* (justifying revision or reconceptualization). When studying probabilistic phenomena, however, it is hard to draw a clear line. If in your data, a certain form occurs 82% of the time, but your MaxEnt analysis predicts that it should occur 80% of the time, is that an analysis that 'works'? What about if your analysis predicts 75%, or even 65%? Does it make a difference if your 82% number comes from a corpus of several million entries, or from a fieldwork dataset with just hundreds? In this chapter, we suggest several methods that can address such questions, which are traditionally included under the term **model fit**. We suggest an approach that is both quantitative (how good is the fit between the MaxEnt analysis and the data?), and qualitative (what mistakes does the MaxEnt analysis make? What forms is it good at, or bad at?).

In the first sections of this chapter, we first cover some basic methods that can help answer questions like the above: first graphing and visualization (§5.2), then statistical testing (§5.3).

5.2 Graphing and visualization

Although formal statistical methods are a gold standard for assessing models, experts also emphasize (e.g. Tukey, 1980; Baayen, 2008: p. x) the need for visualizing and sorting data in a way that engages our intuition and qualitative thinking. Beyond this, there are instances where graphs can reveal essential patterns that are outright missed by standard statistical analysis; two famous instances are "Anscombe's quartet" (Anscombe 1973) and "Simpson's paradox" (Simpson 1951).[28] Lastly, when it comes to presentation of MaxEnt OT work, we suggest that audiences will generally appreciate the chance to apprehend the pattern visually before the statistical analysis is presented to them. Data visualization is of course an extensive field, for which introductory texts include Tufte (2001) and Cairo (2013).

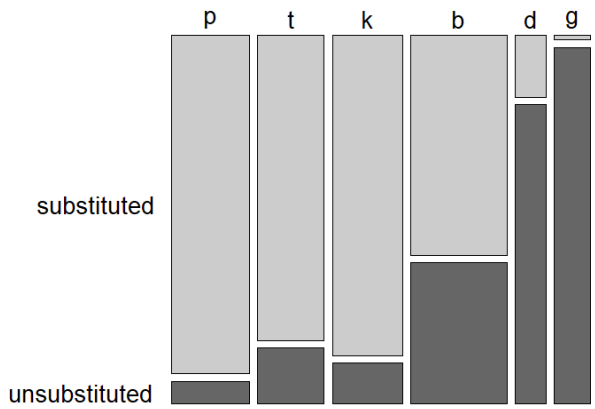
5.2.1 Mosaic plots

Before moving on to discuss the visualization of model fit specifically, we will mention here a type of plot which can be particularly useful to convey the type of data patterns used in MaxEnt

[28] Simpson (1951) makes its point without using graphs, but a helpful visual example of a Simpson-case can be found at this site: https://nastengraph.medium.com/why-you-should-always-visualize-your-data-first-18b8f432bc14, from Anastasiya Kuznetsova.

analysis: the **mosaic plot**. A mosaic plot is a bar graph in which the width of the bars indicates the number of data points contributing to each bar. For example, Figure 26 is a mosaic plot corresponding to the simple bar plot given in Figure 13 to illustrate the data pattern of Tagalog nasal substitution. In Figure 26, the width of each bar conveys the number of forms contributing to each observed substitution rate. The bar for 'b' is thus wider than the others, reflecting the greater number of 'b' forms in the corpus.

Figure 26: A mosaic plot version of Figure 13

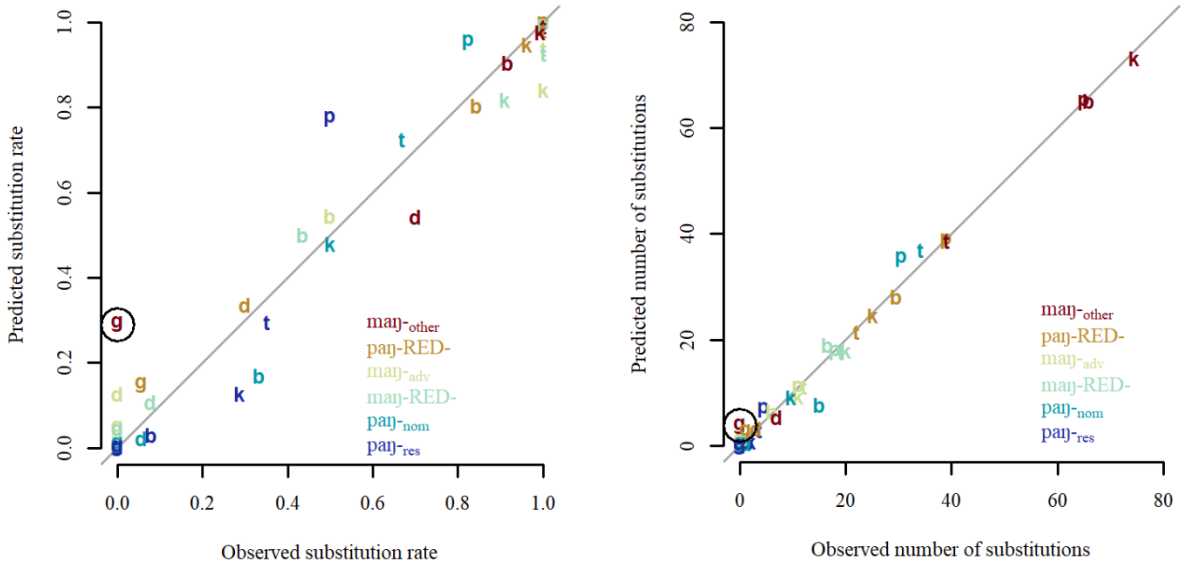


5.2.2 Visualizing model fit using scatterplots

For the purpose of comparing the predictions of a MaxEnt model to the data, we have already suggested using scatterplots, comparing the predicted frequencies of rival candidates with their attested frequencies. To make such scatterplots, we must first resolve a minor issue of scaling. The numbers to be plotted will consist of model probabilities (predictions) and empirical counts (observations). To render them comparable, we must either divide the empirical counts by total counts, converting them to estimated probabilities, as we did for K5 in Figure 4, p. 36; or we multiply the model probabilities by total counts, converting them to predicted counts. For the Tagalog substitution rates modeled in Chapter 2, we show the outcome of both methods compared below in Figure 27.

Figure 27. Two types of scatterplot: fractions vs. counts

a. Observations converted to fractions *b. Model probabilities converted to counts*



In this case, the analysis looks much cleaner using counts rather than probabilities. The reason is that some data types are represented by many tokens, and some by very few in the original data. Observed probabilities based on just a few data points are highly vulnerable to random variation. For instance, the worst-looking outlier point in Figure 27a, circled, is the one at (0, 0.29); measured in terms of counts, this turns out to be a misprediction of just 3.8 cases where zero were observed. On the scale of Figure 27b, this difference appears as a rather modest error — one that could easily be due to sampling error in a data set of this size. More generally, modeling mismatches that likely arise from small numbers will get crowded into the lower left corner of scatterplots like Fig. 27b, even when they loom large in scatterplots like Fig. 27a.

Here are some normative practices for creating scatterplots. First, if the modeling involves just two plausible candidates per input, it makes sense to plot just one of them, as we have done for Tagalog. The remaining points are redundant and would result in a more cluttered scatterplot. Second, it is sensible to scale the axes to be of equal length, since comparable entities are being compared. Third, it is useful to add in the diagonal line $y = x$, indicating the locus where prediction is perfectly accurate; this assists the eye in spotting the data points that deviate most from model accuracy. Lastly, we recommend that observed values be placed on the horizontal axis, and predicted on the vertical axis. This allows for more intuitive evaluation of a point's deviation from the y=x line (we can say that the point "needs to move up" or "needs to move down").[29]

An important use of scatterplots is in trying to improve a tentative analysis, perhaps by adding further constraints. When we look at a scatterplot, the eye's attention is drawn to the

[29] Our advice goes against the published counsel of Piñeiro et al. (2008); their advice is based on the interpretation of slope and intercept values in a linear regression, which is not applicable to probability spaces.

outlier points. These can have various interpretations, concerning which we give our own intuitive judgment. (a) When the observed value is zero and the predicted value substantially above zero, this is a serious failure: in traditional generativist terminology, this is *overgeneration*. (b) Likewise, when the predicted value is zero and the observed value well above zero, this is also a serious failure; i.e. is *undergeneration*. (c) Cases where the data point is well off the diagonal, but not close to zero on either axis, also indicate defects of analysis, though perhaps not as serious, and may conceivably be due to random factors rather than being systematic errors. Obviously, the inspection of outliers from this point of view will be assisted by having your software annotate the points with data labels, as we have done in Figure 27 above. Examining the outliers will often suggest new constraints, which are best chosen (§4.3.1) based on our knowledge of the data pattern, language typology, and previous theoretical work.

Our sigmoid complexes (e.g. Figure 17) can be thought of as another type of scatterplot, one which illuminates the way the analysis derives the observed result.

The search for outliers can also be done non-visually by adding columns to the analytic spreadsheet. In Columns U and V of **Tagalog_testing.xlsx\find_outliers**, we have given the value obtained by subtracting the predicted value (as probability, then as counts) from the observed value. We have autoformatted this column to show up in pink when the error is greater than 0.01 (overgeneration) or less than −0.01 (undergeneration). We also found the worst errors by using the Excel MAX() and MIN() functions.

5.3 Statistical assessment of MaxEnt grammars

Much of modern statistics can be classified as either **hypothesis testing**, or **model selection** (Burnham and Anderson 2002:3). Since our goal is to find an appropriate grammatical analysis of language data, it is the latter approach that is primarily applicable for MaxEnt OT modeling.

5.3.1 Why correlation should not be used to assess model fit

We begin by disposing of one method of model evaluation that may seem obvious but may not actually be appropriate for use — the familiar **correlation coefficient** (*r*; see e.g. Field 2026:ch. 7), as applied to observed vs. predicted probabilities. There are several reasons not to use correlations in this context.

First, the requirements for using correlation in statistical analysis are not met (the *r* statistic assumes that the data are continuous and normally distributed.

Second, in linguistic analysis, we seek a model that matches the observed frequencies exactly — but it is possible for predictions to be perfectly correlated with the data without matching exactly. For example, a set of predictions that fall in the range 0.25 to 0.75 could correlate perfectly with a set of observed values that fall between 0 and 1; the correlation is the maximum of 1, but the model nevertheless is imperfect, for it makes errors as large as 0.25. For a worked-out example of this type, see **PerfectCorrelationFromAnImperfectModel.xlsx**.

Lastly, a high correlation can conceal the effects of a bad analysis, in which the analyst has failed to notice a generalization. In **CorrelationFailsToDetectMissedGeneralization.xlsx** we

give paired analyses for a simple system of vowel devoicing: vowels devoice finally, never in stressed syllables (which are always initial), otherwise only after voiceless consonants (a nonstochastic pattern). The constraints needed, with fitted weights, are given in (81).

(81) *Toy example illustrating how correlation values can be deceptive*

	Constraint	*Characterization*	*Weight*
a.	*VOICELESS STRESSED	Penalize a voiceless vowel in initial syllable	100.0
b.	*VOICED FINAL	Penalize a voiced vowel in word-final position	55.0
c.	AGREE	Penalize a voiced vowel after voiceless consonant	44.9
d.	*VOICELESS VOWEL	Penalize voiceless vowels	22.6

(82) *Sample tableaux: full analysis*

	voicing:	**freq.**	*VOICELESS STRESS 100.0	*VOICED FINAL 52.0	AGREE 44.9	*VOICELESS VOWEL 22.6	*P*
/ˈpipipi/	+ + +			1	3		
	+ + −				2	1	
	+ − +			1	2	1	
	+ − −	1			1	2	~1
	− + +		1	1	2	1	
	− + −		1		1	2	
	− − +		1	1	1	2	
	− − −		1			3	
/ˈbibibi/	+ + +			1			
	+ + −	1				1	~1
	+ − +			1		1	
	+ − −					2	
	(etc.)		1	1		1	

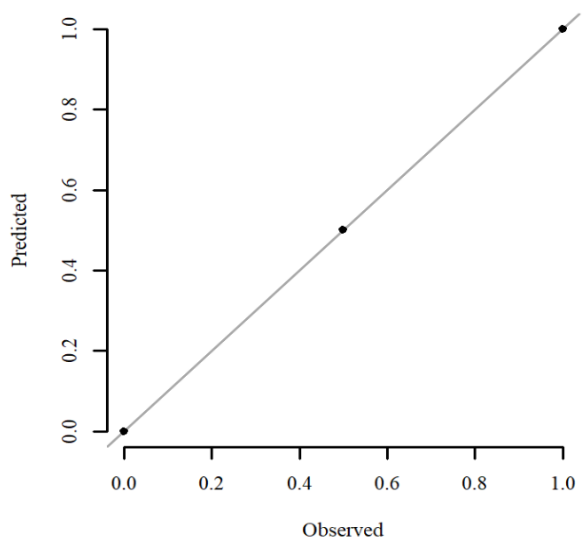
This analysis turns out to match the data almost perfectly, and its value for *r* accordingly is 1. This full analysis may be compared with a problematic alternative in which a careless linguist has failed to notice the post-voiceless generalization embodied in (81c), and therefore has not included AGREE. For this analysis, the tableaux entries are distinguished solely by syllable count, and forms like [ˈpipi̥bi̥] and [ˈpibibi̥] act as members of a lexically varying population, much as in Tagalog /pa**ŋ-p**igha ˈtiʔ/ → /pa**m**igha ˈtiʔ/ vs. /pa**ŋ-p**o ˈʔok/ → [pa**mp**o ˈʔok], given earlier in (39a). This miscontrual of the data is readily modeled in MaxEnt, as seen in (83).

(83) *A tableau for the grammar that is missing* AGREE

			*VOICELESS STRESS	*VOICED FINAL	*VOICELESS VOWEL	
	voicing:	freq.	100.0	39.6	0	*p*
'σ σ σ	+ + +			1		
	+ + −	0.5			1	0.5
	+ − +			1	1	
	+ − −	0.5			2	0.5
	− + +		1	1	1	
	− + −		1		2	
	− − +		1	1	2	
	− − −		1		3	

This analysis, while obviously wrong, also has a correlation of 1. In the scatterplot below, the data point at (0.5, 0.5) represents the second and fourth tableau rows of (83); i.e. the outputs that the bad analysis wrongly attributes to lexical variation.

Figure 28: A deceptive scatterplot: pooled predictable cases treated as random variation



This example, while intended to illustrate a point about correlation, also embodies a point that any linguist looking at variation needs to keep in mind: there always a risk of assuming that data reflect variation when in fact they reflect yet-undiscovered generalizations.

5.3.2 Assessing models with log likelihood

Having disposed of correlation, we now turn to a statistic that is more trustable than correlation for evaluating MaxEnt models: it is the very same log likelihood statistic that we have been using to set the constraint weights. In the present context we can note that in the good analysis of the toy example above (i.e. CorrelationFailsToDetectMissedGeneralization.xlsx) has

a log likelihood of -1.46×10^{-9} (essentially zero, the best possible value); and the shortcomings of the bad analysis are reflected in its log likelihood, which is −5.54. In our other illustrative example, PerfectCorrelationFromAnImperfectModel.xlsx, we calculate that the model that achieves a perfect match receives a much higher log likelihood (−440.4) than the model that that only achieves perfect correlation (−531.6).

Log likelihood can be used as a simple and intuitive way to compare different analyses of a dataset. Often a more appropriate comparison is to use not log likelihood per se, but a statistic derived from log likelihood, such as the Akaike Information Citerion, calculated as $2 \times \mathrm{LL} - 2 \times n$, where n is the number of constraints. The AIC appropriately penalizes analyses that achieve high log likelihood through the use of many constraints. For extensive discussion of the use of the AIC in modeling, see Burnham and Anderson (2002).

5.3.3 Assessing the usefulness of particular constraints I: zero weights

We turn next to the statistical assessment of the individual constraints that comprise the analysis.

In many cases, we first learn that a constraint is of no help in an analysis when we fit the weights and find that the best-fit weight of our constraint is zero. We discussed zero weights earlier in §4.4, covering cases where the zero arises from multiple equally-effective weightings, or reflects failure to locate appropriate inputs and candidates. Here we address zero weights more akin to what we studied in (61), where the zero is meaningful (if the weight is set any higher the analysis becomes less accurate). In such cases, zero weight might be taken as a clear indication that the constraint does not belong in the grammar.[30]

When a constraint receives a best-fit weight of zero, it often emerges that the constraint is useless because its work is better done by others constraints already present in the grammar. To give one example, for the Turkish vowel harmony grammar of Chapter 6 (see (93c)) we find that if we add to the analysis the (plausible-seeming) constraint $*\begin{bmatrix}-\text{back}\\+\text{round}\end{bmatrix}\begin{bmatrix}+\text{back}\\-\text{round}\end{bmatrix}$ ("avoid front rounded followed by back rounded), it is fitted with a weight of zero. As it turns out, the constraints of the existing analysis already do all of the work of this constraint.

The usual reason that the weight for a useless constraint comes out *exactly* zero is that the analyst has chosen to forbid negative weights. For instance, if we re-weight the augmented Turkish constraint set with negative weights allowed, we find that the true best-fit weight of our useless constraint is a tiny negative number: −0.07.[31] The random fluctuations in almost any real data set mean that non-meaningful constraints get fitted with a weight of not exactly zero, but close to it; these get truncated to zero when negative.

[30] If the constraint set is taken to be universal, the question of removing constraints does not arise; in this section we will assume parochial constraints.

[31] We remind the reader of table (56), p. 72, intended to assist intuition about weight magnitudes.

5.3.4 Assessing the usefulness of particular constraints II: the Likelihood Ratio Test

Even among constraints that receive positive weights, skepticism is often still appropriate. A constraint can have a non-zero weight and yet serve no useful purpose in the analysis. Often such constraints are redundant, as discussed in the previous section; even high-weighted constraints can be redundant if they cover very few data. In this section, we present a statistical test, the **likelihood ratio test** (Wasserman 2004), that can serve to detect such cases.

The likelihood ratio test is based on model comparison: we consider a full model that includes the constraint ℂ we are interested in, and compare it with a minimally different model that omits ℂ. The minimal comparison serves as an assessment of the contribution that ℂ makes to the model fit. Specifically, we want to know whether the improvement in log likelihood yielded by including ℂ is "worth" the addition of the extra constraint to the analysis. The procedure is given in (84).

(84) *Testing constraint* ℂ *for significance with the Likelihood Ratio Test*

a. Calculate the log likelihood of the data under a MaxEnt grammar containing ℂ, with optimized weights.
b. Form a second grammar by deleting ℂ from the first grammar, and reoptimize the weights.
c. Subtract the log likelihood of the second grammar from the log likelihood of the first, yielding a positive number x.
d. Multiply x by two: this value follows a chi square distribution.
e. Look up the p-value associated with $2x$ in the chi square distribution, with one degree of freedom (because one constraint was removed).

The test yields a p value, indicating the probability that the log likelihoods would differ as they do if the constraint ℂ was actually useful in analyzing the data. Low values (a threshold can be assigned, such as 0.001) can be taken to mean that the observed data are much more likely under the analysis that contains ℂ than under the analysis that does not. High values indicate that the data is just as likely or nearly as likely without ℂ as with it. For discussion of the concept of p-values, and the perils associated with them, see Field (2026:ch. 3).

In **Tagalog_testing.xlsx**, we illustrate how this computation can be carried out for the model of Tagalog nasal substitution presented in Chapter 3. Example (85) gives excerpts from the crucial columns. The boldface values in Column W were all computed separately and pasted in as values (so they wouldn't change on subsequent runs). On the right side, Column W is repeated, showing how each cell was calculated.

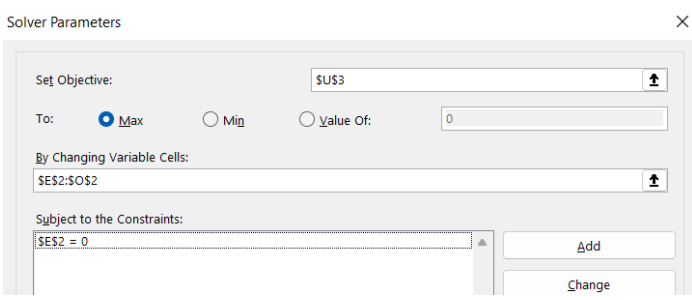
(85) *Application of the Likelihood Ratio Test to the Tagalog analysis*

	W	X
3	**-248.61**	**Full model**
4		
5	**-296.13**	**Omit NASSUB**
6	47.52	difference
7	95.04	twice difference
8	1.87E-22	*p*
9		
10	**-428.67**	**Omit *NC**
11	180.06	difference
12	360.12	twice difference
13	2.65E-80	*p*
14		
15	**-273.62**	**Omit *[n**
16	25.01	difference
17	50.02	twice difference
18	1.52E-12	*p*

Column W as formulae
(calculated in U3, then pasted)
(calculated in U3, then pasted)
=W$3-W5
=2*W6
=CHIDIST(W7,1)
(calculated in U3, then pasted)
=W$3-W10
=2*W11
=CHIDIST(W12,1)
(calculated in U3, then pasted)
=W$3-W15
=2*W16
=CHIDIST(W17,1)

Cell W3 gives the log likelihood obtained when all 11 constraints of the full analysis are employed, as calculated with the Solver; this is about −248.61. W5 gives the log likelihood obtained when we remove NASSUB from the grammar. This calculation is most easily done by using the Solver Parameters box to require that the weight of the tested constraint be zero; under MaxEnt this is the same as removing it. The image in Figure 29 shows what this box looks like when we impose this condition (in the full spreadsheet, E2 is the cell containing the weight of NASSUB):

Figure 29: Solver Parameters box: zeroing out weight of NASSUB



The log likelihood value returned by this calculation, −296.13, was pasted into cell W5, as seen above in (85). Next, cell W6 of (85) performs the subtraction needed to assess the improvement in log likelihood created when NASSUB is included in the grammar, namely (−248.6) − (−296.1) = 47.52. Cell W7 doubles this value, obtaining 95.04. Lastly, cell W8 uses the CHIDIST() function to return a *p*-value, 1.87×10^{-22}. Performing the same procedure for all the constraints, we obtain the values in (86) (*ΔLL* denotes change in log likelihood).

(86) *Significance testing by Likelihood Ratio Test of the Tagalog nasal substitution analysis*

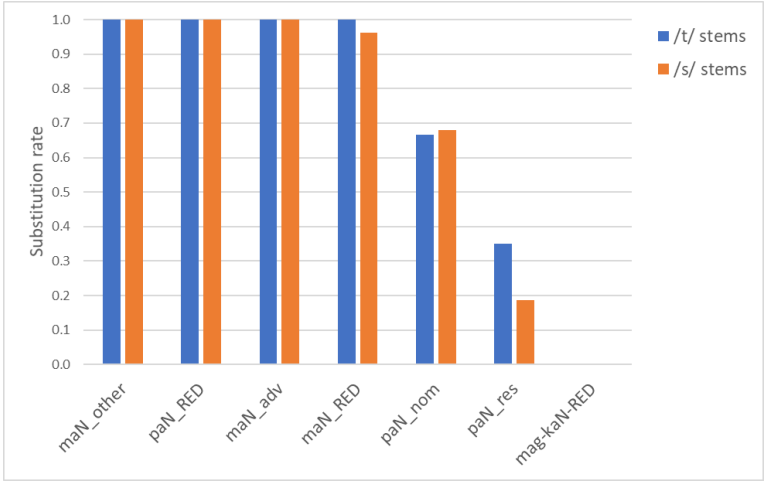
Constraint	*ΔLL*	*Weight*	*p*
NASSUB	47.52	2.28	1.87×10^{-22}
*NC	180.06	4.65	2.65×10^{-80}
*[n	25.01	2.10	1.52×10^{-12}
*[ŋ	54.04	3.16	2.59×10^{-25}
UNIF-/maŋ-$_{other}$/	0	0	1
UNIF-/paŋ-RED-/	2.24	0.85	0.034
UNIF-/maŋ-$_{adv}$/	9.95	2.09	8.18×10^{-06}
UNIF-/maŋ-RED-/	19.70	2.27	3.46×10^{-10}
UNIF-/paŋ-$_{noun}$/	75.63	3.85	9.15×10^{-35}
UNIF-/paŋ-$_{res}$/	73.51	5.69	7.77×10^{-34}
UNIF-/mag-kaŋ-RED-	257.60	15.15	4.66×10^{-114}

We remark briefly on these values. For UNIF-/maŋ-$_{other}$/, the *p* value of 1 is expected; its weight is zero for a mathematical reason (§4.4.2), and a zero-weighted constraint cannot make any difference to the log likelihood of the grammar. For UNIF-/paŋ-RED-/, we obtained a *p* value of 0.034. This makes only a weak case that the constraint is useful; were we to omit UNIF-/paŋ-RED-/, forms with the /paŋ-RED-/ prefix would be expected to behave just like forms with /maŋ-$_{other}$/ – which is not far from the truth. The other values look very impressive, but it is important to remember that *p*-values don't mean "probability that the analysis isn't true" – they only mean the probability that the weight could be as high as it is by mere accident. Most proposed analyses compete with alternative analyses – which might also receive impressive *p*-values.

5.3.5 An example of failing the Likelihood Ratio Test

Consider next an instance in which the likelihood ratio test strongly suggests that a constraint is not appropriate for inclusion in the grammar, even though it bears a positive weight. This arises when we expand the dataset for Tagalog to include stems starting with /s/, as in /ma**ŋ**-ˈ**s**ulat/ 'gloss' → [ma-**n**uˈlat] 'to write professionally'. In the original research by Zuraw, the /s/ stems are lumped together with the /t/ stems, for good reason. At the theoretical level, the constraint set employed does not distinguish /t/ and /s/ stems, as any voiceless alveolar obstruent would yield [n] as its mutated form; and the effects of underlying consonant are modeled at the surface level with the constraints *[n and *[ŋ. In addition, the data pattern for stems beginning with /t/ and /s/ is strikingly similar. Figure 30 shows the substitution rates for these two consonants with all seven of the prefixes studied here.

Figure 30: Substitution rates for /t/ and /s/ stems across seven Tagalog prefix types



The one case where the two columns for /t/ and /s/ diverge in any appreciable way is for the prefix /paŋ-$_{res}$/, where the data in Zuraw's corpus are sparse — only 10 /t/ stems and 8 /s/ stems. So the idea of employing a grammar that treats /t/ and /s/ stems the same is very appealing here.

Just for pedagogical purposes, however, we instead analyzed a synthesized data set, in which the number of instances of nasal substitution for /t/ is artificially boosted by a factor of 1.5; this increases the chance that a /t/-/s/ difference might be statistically confirmed. We also needed to revise the analysis so that it makes a distinction beween /t/ and /s/ stems; i.e. a constraint that penalizes nasal substitution only when the source segments are /ŋ+s/, not /ŋ+t/. We will not dwell on what this constraint might be, and will simply name it literally as *[n] FROM /s/; it penalizes all UR-SR correspondences of the form [ŋs] ~ [n].

In the spreadsheet **TagalogWithSDistinct.xlsx** we introduce *[n] FROM /s/ into a grammar that retains all of the previous constraints. Setting the weights for this new grammar in Excel, we found that *[n] FROM /s/ received a very modest weight, 0.25. Including it improves the likelihood of the grammar from −292.749 to −292.501, a difference of only 0.247, which corresponds to a very unimpressive *p*-value of 0.48. Such a result might well be greeted with a sense of relief: it means we can drop from the grammar a rather complex and ill-justified constraint, secure in the knowledge that whatever it did for us could very easily have been the result of the particular sample of Tagalog that Zuraw was able to extract in her corpus.

C:\Dropbox\AR\MaxEntBook\PrepareSims\Tagalog\TagalogFullWithSDistinct.xlsx

5.3.6 Likelihood ratio testing in R

The practice of checking each constraint for significance with the Likelihood Ratio Test is one in which one can proceed more efficiently using the `maxent.ot` package in R, mentioned above in §2.11. In particular, you can set up a program loop to test the constraints one by one, start the program, and then go play with your cat. A sample script for this purpose can be found in the supplementary materials: **likelihood_ratio_test_loop.R**.

5.3.7 Cross-validation

Large models, especially in computer science, are often checked by the method of **cross-validation**. The goal is to see if they are "overfitted," meaning that so many constraints have been included that the model is achieving an artificially inflated accuracy by fitting random fluctuations in the data. The method works as in (87).

(87) *N-fold cross validation (n = 5)*

a. Divide the data at random into five sets.
b. Omit Set 1, and refit the weights to the data of Sets 2-5.
c. Find the predictions of the refitted model for the data of Set 1.
d. Repeat steps b-c for Sets 2-5.
e. Combine the new predictions for the five subsets and calculate log likelihood, comparing with the original value.

The idea behind this scheme is that in no case is the prediction for a particular data point obtained by using that data point in the training data, so the data don't get to "fit themselves." It can be appropriate to report these cross-validated weights and their associated log likelihood, instead of results obtained by fitting the whole data set at once; this is done by Wilson and Gallagher (2018) in their MaxEnt analysis of Quechua phonotactics.

We suggest using cross-validation to check analyses that use large numbers of constraints, especially morpheme specific ones. However, when a model consists of just a few highly general constraints, we follow the view of Goldwater and Johnson (2003:§4.3), who suggest that cross-validation is unnecessary in such cases: the generality of the constraints already assures that the model cannot be overfitted. As they put it: "the learning algorithm doesn't see words, it only sees patterns of violations," and these patterns are often rather few.

For extensive discussion of cross validation, including instructions on how to perform it in the `maxent.ot` R package, see Mayer et al. (2024).

5.4 On model-building

Generative grammarians have for decades, without using the term, been **model-builders**: they seek an abstract and formalized structure that plausibly represents the internalized knowledge (as diagnosed in various ways) of the speaker. In the context of MaxEnt, it seems helpful to consider issues of model-building as they arise more generally in the sciences. A useful text on model building which we have consulted is Burnham and Anderson (2002).

A key aspect of modeling is that we are seeking not so much accuracy as understanding. Just as a perfect map of a region would be an exact full-size copy of the region itself, and therefore completely useless as a map (Korzybski, 1933), models of linguistic phenomena are useful not because they are perfectly accurate, but because they represent an understanding of the data that has been simplified in some way so as to provide a meaningful answer to a question which the researcher raises. The goal of the analyst, therefore, is not necessarily a constraint set which

perfectly matches a given dataset, but rather a constraint set which answers (or comes as close as possible to answering) a specific research question.

5.4.1 Building a model from a set of potential constraints

A grammatical model submitted for publication often emerges from a gradual process of model-building, with iterated improvement of tentative models. Often, comparisons of the favored model with lesser-performing alternatives is informative. In this section we describe ways of building up an effective model, starting with a set of candidate constraints. The key element is using the likelihood ratio test to guide model construction.

A good starting point for building up a grammatical model is to consider a **constraint space** consisting of the constraints that might in principle be applicable to the problem at hand. This space can be relatively restricted and theoretically informed: for example, containing for example all previously proposed constraints pertaining to a specific problem (as in Hayes et al. 2012), or it can be expansive and exploratory, as in Hayes and Wilson (2008), who consider all strings of feature matrices up to a length of three, or Moreton et al. (2017) who consider all possible conjunctions of up to eight features.

The selection of constraints from a constraint space must be informed by both the specific data at hand and the analyst's knowledge of theory. Algorithms exist (§4.5) to determine the best weights for a set of constraints and a data set, but no algorithm exists that can reliably determine *which* constraints should be considered in the first place. This is true not just for constraints in MaxEnt, but for parameters in any model. Because we are investigating questions whose answers are not yet known, we have only clues regarding what factors might ultimately be relevant, and we must always entertain the possibility that we might be missing an important one.

For MaxEnt analyses specifically, we recommend beginning with a thoughtful pre-theoretic understanding of the data at hand, particularly of what are the crucial generalizations and their possible causes. Having done this, the next step is to consult existing theory. Analytic work in linguistics normally adopts an explicit theoretical framework, which in the context of OT/MaxEnt will include specific hypotheses, grounded in typological study, about what a constraint can be. Indeed, the broader task of theory-comparison can be made more rigorous by trying multiple frameworks and letting each one dictate its own constraint space.

On the other hand, it is hardly a signed and sealed research result that all languages can be analyzed with constraints are already part of some typologically-supported set, and in some cases we must include parochial constraints simply to come anywhere near describing the pattern of the language data. The role of general versus language-specific constraints is likely to remain an open research question for some time; see §7.4 below.

5.4.2 Selecting constraints from a constraint space

Once a constraint space has been defined, then model selection takes a more concrete form: what subset of the constraints in the constraint space should be selected? We seek a set of constraints which are all useful in analyzing the data — each of which improve the model's fit over a model without that constraint. A sensible concrete goal is for the grammar we choose

(constraint set and weightings) to be one in which (a) every constraint tests as significant, by our chosen criterion; (b) excluding grammars ruled out by (a), log likelihood is maximized.

Finding such a grammar may be a daunting challenge when the constraint space is anything other than quite small. This is because of the problem of combinatorics. If the full number of constraints being considered is *n*, then the number of constraint *sets* we would consider deploying as a grammar is 2^n. This can be a very large number for values of *n* commonly used in linguistics, so finding the best-performing possible grammar in which every constraint tests as significant may be beyond our computational capacity.

The problem is indeed not reliably solvable, to our knowledge, but there are two heuristic methods (Walker, 2010:39) that can help to find something close to the best solution. These heuristics have different names in different sources.

Bottom-up method. Given a set of constraints { C_1, C_2, … C_n }, choose the one-constraint grammar { C_i } that, when optimally weighted, bears the highest log likelihood. Then keep adding new constraints, always picking the constraint that creates the greatest improvement in log likelihood. When there is no added constraint creates a significant improvement in log likelihood, terminate and keep the already-chosen constraints as the final grammar.

Top-down method. Given constraints { C_1, C_2, … C_n }, form the grammar consisting of all the constraints, suitably weighted. Next, find the constraint that produces the smallest improvement in log likelihood and, *if* it fails to pass the likelihood ratio test, delete it, forming a smaller grammar. Continue until a grammar is obtained in which every constraint passes the likelihood ratio test.

In Hayes et al. (2012), both methods are applied to a constraint space with 88-constraints. Somewhat reassuringly, the grammars thus obtained are rather similar.

For discussion of potential pitfalls with the bottom-up and top-down methods see Field (2026:§8.10.1). Of the two methods, Field (p. 1039) recommends top-down for the avoidance of "suppressor effects", in which a constraint that might have been helpful fails to make it in by accidents of the order of addition.

A recurring theme in selecting constraints from a constraint space is the important role of theory and previous research results. For instance, Field (p. 475) suggest that if one is using a bottom-up method, then "known predictors (from other research) into the model first". In the present context, this implies first adding the constraints that have a solid background in theory and typology, and only then the parochial constraints. Burnham and Anderson (2002) likewise lay heavy emphasis on favoring theoretically-motivated principles, and not letting a statistical search "fish" at random among a large set of dubious factors.

In any case, both Walker and Field recommend reporting the full constraint space in journal publication.

5.5 On obtaining probabilistic data

We conclude the chapter with advice about obtaining the probabilistic data employed in MaxEnt analysis. The spreadsheets used in Chapters 2 and 3 were in fact the end point of extensive research and labor. Marshalling the data needed for a MaxEnt analysis is likely to involve considerable expertise in the language(s) of interest and in data collection, possibly including experimentation. This section offers a brief introduction to some of the available techniques, providing citations to sources from which in-depth training can be obtained.

5.5.1 Lexical corpora

We note up front that there exists a whole field of corpus linguistics, covered extensively for instance in Durand et al. (2014). We briefly cover a few types of corpora here.

For studies of gradience in the lexicon, as for instance in Chapter 3, it often suffices to analyze a lexical corpus, i.e. a list of words. Web search can help you with finding such corpora, often in the form of online dictionaries, such as the various Wiktionaries at www.wiktionary.org. In other cases, the only available list of words will be a hard-copy, printed dictionary. Hard-copy dictionaries can be reduced to digital form by optical character recognition ("OCR"), though further processing will be needed to isolate the entries from the definitions. For example, one such dictionary-based corpus, used for Shona phonotactics by Hayes and Wilson (2008), was scanned by the CBOLD project (http://www.cbold.ddl.cnrs.fr) from the print dictionary of Hannan (1959). An alternative to OCR, which for shorter dictionaries might be faster, is simply to use patience and hand-type every entry (e.g. Hayes and Wilson 2008, typing Dixon 1981), perhaps sampling every tenth (or n^{th}) entry.

The more ambitious dictionaries tend to include great numbers of words not known to the average native speaker, particularly in languages subject to rapid cultural change. Here, an effective control can be to ask/employ a native speaker to go through a dictionary and note which forms are familiar to him/her. This was the strategy, for instance, of the Turkish Electronic Living Lexicon (https://linguistics.berkeley.edu/TELL/TELLhome.html), used in Chapter 6 below for our analysis of vowel harmony in Turkish stems. The TELL speakers were familiar with about 17,500 of the 30,000 dictionary words that were submitted to their inspection.

The vowel harmony data for Hungarian in Hayes and Londe (2006) were also scrutinized individually by native speakers. These data involved 1130 stems, each notated for their harmonic class (back or front) by coauthor Londe, along with two other native speakers recruited by Dr. Londe. In the present day, many native speaker consultants are familiar with Excel and can patiently fill in appropriate blank columns with the forms used in their own speech.

When a dictionary provides no pronunciation entries, a phonologist will often face the need to convert an alphabetic orthography into phonemic transcription. For some orthographic systems, there exist downloadable conversion system, or it can be done with a modest amount of user-created code. Suitable attention must be paid to digraphs (which create parsing problems; *ph* in *chophouse*) or neutralization of lost contrasts still spelled (Spanish *ll* and *y* are both /j/). In languages like English or French, with historical spelling, conversion by code is unreliable and can at best provide a rough draft for later hand-editing. An alternative to conversion is dictionary

look-up, which however can be defeated by homographs (*lead*, *annex*); again, hand-editing will be required.

Often, lexical corpora are **lemmatized**, meaning that the inflected forms of a single stem are grouped together This permits, for instance, a study of the phonotactics that hold specifically of stems, as for Turkish in Chapter 6 below.

For many phonological applications, such as study of alternations, we need to study complete paradigms. Depending on the type of paradigm needed, dictionaries sometimes supply enough information to get started — listing out exceptional paradigm members with their stems, for example. Pedagogical materials for the language can be helpful, such as the books of the *501* [*Language X*] *Verbs* series published by Simon and Schuster.

For some languages, though, the paradigm of any stem is so large that listing all the forms is impossible — for instance, Odden (1981:17) estimated that the verb paradigm of Shona includes at least 16 trillion forms. In such cases we need to reduce the data into a form that we can cope with. For well-studied languages, we can turn to the method of **principal parts**, a technique widely used in language pedagogy. These are particular inflected forms that, if we know them, we can use them to predict all of the other inflected forms. For languages (e.g. Latin) where the principal parts can be shown to be reliable, we can focus our attention on just the principal parts, knowing that we will have in principle covered all of the data. For theoretical development of the notion of principal parts, see e.g. Finkel and Stump (2007) and Bonami and Boyé (2002).

For some purposes it may be helpful to *synthesize* full paradigms using software. This has the advantage of providing ample data, but of course runs the risk of missing irregular forms. For work with synthesized paradigms, see Hayes and Wilson (2008:413) and Cotterell et al. (2017).

5.5.2 Text and audio corpora

Text corpora can be gathered from books, newspapers, websites, and other written sources, or from transcripts of audio, such as transcripts of speeches or podcasts, or movie and television subtitles (e.g. Brysbaert and New, 2009). These can be useful to the phonologist for establishing usage frequencies of specific words and stems, but also for collecting data on phonological processes that are reflected in the language's orthography. An example of such a pattern we have already seen is Tagalog nasal substitution, whose study was based on Zuraw's text corpus.

Audio corpora are of course more unwieldy, and tend to include fewer utterances than a text corpus. They also vary in terms of how closely they are transcribed — for example, the Buckeye Corpus (https://buckeyecorpus.osu.edu) contains phonetic transcriptions, but many corpora contain only orthographic transcriptions, as, for example, the TED talks used by Fuller (Chapter 2) for her Khalkha data. A helpful technique, used by Fuller, is to use the orthographic annotations to find instances of the phonological configuration of interest, and then consult the sound file to ascertain whether the process of interest applies.

Corpora for a wide variety of languages can be found online. Sites that host large collections of corpora can be a good place to start, such as the Endangered Languages Archive (https://elararchive.org/), the Hamburg Centre for Language Corpora (https://www.fdr.uni-

hamburg.de/communities/hzsk/), or the Linguistic Data Consortium (https://www.ldc.upenn.edu/). Corpora vary widely in how they have been constructed, annotated, and cleaned (inspected for typos, non-word elements, uninterpreted numerals, and so on). It is important to carefully inspect a corpus's documentation to know these details.

Often, no corpus exists, especially for less well studied languages. A technique available in this case is **web scraping**, the process of extracting linguistic material by machine from Internet sources. Here, it is useful for one's scraping software to be able to identify the language of a web site, something which is in fact generally possible with a reasonable sample. For example, in Chapter 3 we noted that much of the data for Kie Zuraw's studies of Tagalog nasal substitution were gathered using a web-scraper that identified Tagalog websites by using a short list of known Tagalog words. Mayer (2021) improved his data quality (we suspect) for a study of Uyghur vowel harmony by limiting his search to the sites of two leading Uyghur-language newspapers; he found that his work was aided by already-programmed Uyghur scrapers which he could adapt to the task.

Text corpora can be reduced to lexical corpora by machine-collating the included words. The collation process will also yield frequency estimates. Creation of lexical corpora from text corpora may impose an accuracy cost, however, from combining the entries for homographs.

Lastly, corpora will vary in *style* depending on their source. If one can access bodies of archived speech, these will likely embody the vernacular style of the language, which may be quite different from what is used in books and newspapers. Jo (2024b) found quite different patterns of vowel harmony in Korean by examining corpora from various genres (historical Korean, newspapers, spontaneous speech, wug tests); these data provide a clear diachronic picture of a phonological process gradually losing its productivity.

Once we are in the domain of style, we have ventured into the territory of sociolinguistics, a field which has long excelled in the quantitative study of corpus data; for references, see §2.2.

5.5.3 Field notes

For many languages, research by linguists is at an early stage. For such languages, work of the kind described here — or indeed, any sort of theoretical linguistics — cannot be done until a firm descriptive foundation has been achieved. This work will include establishing the system of contrasting phonemes and tonemes, compilation of a rudimentary dictionary, coverage of the system of inflectional contrasts, and coverage of the basics of word order and other syntactic structure. For fieldworkers, this foundational work is not only laborious but frequently poses its own intellectual puzzles and challenges, even before any issues of current theory get addressed.

Once a certain body of descriptive work is complete, it begins to be possible to study gradience. Suppose, for instance, that one were to compile all of the data files possessed by a single investigator. The data are likely to involve multiple areas of the grammar, but the words collected for diverse purposes may independently be usable for study of phonology. Thus, the data used by McPherson for her Tommo So reference grammar (McPherson 2013) and other works proved sufficient for a detailed quantitative study, including MaxEnt analysis, of the Tommo So vowel harmony system (McPherson and Hayes 2016).

5.5.4 Experimentation

The gathering of linguistic data from experimentation is a vast field, with considerable methodological content. Much of this content addresses gradience, for gradience is indeed the normal outcome for almost any experiment on language whose design permits it to be detected. For students interested in pursuing work in linguistic experimentation, we recommend coursework if possible, but some useful books to consult would include Solé et al. (2007), and Podesva and Sharma (2013). For the case of phonology, we can taxonomize the widely-used experiment types as follows.

An experiment that asks participants to provide inflected forms for novel stems is called a **wug test**, after an item from Berko's (1958) landmark study: the hypothetical noun *wug* [wʌg] is likely to have the plural *wugs* [wʌgz]. A wug test experiment tests for the productivity of both the morphological operation involved (here, English plural suffixation) and of whatever phonology is called forth by the relevant morphological operation (here, voicing agreement between stem and affix). The stimuli in a wug test can be either novel, linguist-created forms like *wug*, or existing stems that happen not to possess inflected forms of the kind being sought (e.g. for many English speakers the past participle of *stride*); the latter may be a more natural task for speakers (Kuo 2023). Wug tests can also vary in whether they ask participants to volunteer a form, as in "Please tell me the past tense of *spling*," or to rate a form, as in "Please rate on a scale of 1 to 7 what you think of *splung* [splʌŋ] as the past tense of *spling*." Some publications on wug-testing that include implicit advice for practitioners include Moreton and Pertsova (2024) and Zuraw and Hayes (2017).

Another major type of test is the submission of novel forms to native speakers for ratings of phonotactic well-formedness, as in for example Greenberg and Jenkins (1964), Scholes (1966), Daland et al. (2011), and many others; see Chapter 6 below. While this kind of test is sometimes also called a wug test, we feel it is more precise to use the alternative term **blick test**, which derives from a thought-experiment in Chomsky and Halle (1965), who surmised (rightly, we think), that English speakers would regard the nonce form *blick* [blɪk] as well-formed, comparable to existing *brick* [bɹɪk], but *bnick* *[bnɪk] as phonotactically impossible.

Both wug and blick tests can be carried out either on the real language spoken by the experimental participants, or on an imaginary, made-up language that they must learn on the spot; the latter are called Artificial Grammar Learning experiments (AGL; Finley 2017, Culbertson 2023). The two methods balance the advantages of naturalness/familiarity with the ability of the investigator to exercise full control over the linguistic pattern presented.

5.5.5 Gathering data from multiple speakers

Probabilistic grammars, as with other work in generative grammar, are meant to model the internal linguistic system of a single speaker. When we are dealing with variation, though, variation in many datasets can arise both from within speakers and across speakers, and probably most often a mixture of both. Ideally, MaxEnt analyses should only analyze the variation that is within speakers. For this reason, it is important to check whether the variation in a dataset is between or within speakers.

This problem is a difficult one, but we offer a few suggestions. First, understanding the linguistic background of one's speakers is crucial. In finding consultants for fieldwork, it is often possible to ensure that the multiple speakers being studied speak closely similar varieties; experimenters often include a language background questionnaire as part of the participants' task and discard post hoc the responses not coming from speakers of the target dialect. Corpora may contain speaker labels along with demographic information. Once data have been gathered, it helpful to scrutinize them in order to detect and systematize any differences between speakers that are found. Finally, in both testing and elicitation it is important to give each speaker multiple examples that are structurally parallel; to the extent that individual speakers vary in their responses to these parallel stimuli, we have evidence that we are seeing speaker-internal variation, not dialect variation.

Bibliography

Albright, Adam. 2002. "Islands of Reliability for Regular Morphology: Evidence from Italian." *Language* 78: 684–709.

Albright, Adam. 2008. "Explaining Universal Tendencies and Language Particulars in Analogical Change." In *Linguistic Universals and Language Change*, edited by Jeff Good. Oxford University Press.

Albright, Adam, and Bruce Hayes. 2003. "Rules vs. Analogy in English Past Tenses: A Computational/Experimental Study." *Cognition* 90: 119–61. https://doi.org/10.1016/S0010-0277(03)00146-X.

Alderete, John D. n.d. "Structural Disparities in Navajo Word Domains: A Case for LEXCAT-FAITHFULNESS." *The Linguistic Review* 20 (2–4): 111–57. https://doi.org/10.1515/tlir.2003.006.

Alderete, John, and Sarah Finley. 2023. "Probabilistic Phonology: A Review of Theoretical Perspectives, Applications, and Problems." *Language and Linguistics* 24 (4): 565–610.

Anderson, Gregory D. S. 2013. "The Velar Nasal." In *WALS Online*, edited by Matthew S. Dryer and Martin Haspelmath, v2020.4. Zenodo. https://doi.org/10.5281/zenodo.13950591.

Anscombe, F. J. 1973. "Graphs in Statistical Analysis." *The American Statistician* 27 (1): 17–21.

Anttila, Arto. 1995. "How to Recognise Subjects in English." In *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*, edited by Fred Karlsson, Atro Voutilainen, Juha Heikkilä, and Arto Anttila. Mouton de Gruyter.

Anttila, Arto. 1997. "Deriving Variation from Grammar: A Study of Finnish Genitives." In *Variation, Change, and Phonological Theory*, edited by Frans Hinskens, Roeland van Hout, and W. Leo Wetzels. John Benjamins Publishing Company.

Anttila, Arto, and Giorgio Magri. 2018. "Does MaxEnt Overgenerate? Implicational Universals in Maximum Entropy Grammar." *Proceedings of the 2017 Annual Meeting on Phonology* 5. https://doi.org/https://doi.org/10.3765/amp.v5i0.4260.

Baayen, Harald, Robert Schreuder, Nivja de Jong, and Andrea Krott. 2002. "Dutch Inflection: The Rules That Prove the Exception." In *Storage and Computation in the Language Faculty*, edited by Sieb Nooteboom, Fred Weerman, and Frank Wijnen. Springer.

Baayen, R. Harald. 2008. *Analyzing Linguistic Data: A Practical Introduction to Statistics Using R*. Cambridge University Press.

Bayley, Robert. 2002. "The Quantitative Paradigm." In *The Handbook of Language Variation and Change*, 1st ed., edited by J. K. Chambers, Peter Trudgill, and Natalie Schilling-Estes. Blackwell Publishers.

Becker, Michael. 2009. "Phonological Trends in the Lexicon: The Role of Constraints." PhD Thesis, University of Massachusetts Amherst.

Becker, Michael, Nihan Ketrez, and Andrew Nevins. 2011. "The Surfeit of the Stimulus: Analytic Biases Filter Lexical Statistics in Turkish Laryngeal Alternations." *Language* 87 (1): 84–125. https://doi.org/10.1353/lan.2011.0016.

Becker, Michael, Andrew Nevins, and Jonathan Levine. 2012. "Asymmetries in Generalizing Alternations to and from Initial Syllables." *Language* 88 (2): 231–68.

Beckman, Jill N. 1997. "Positional Faithfulness, Positional Neutralisation and Shona Vowel Harmony." *Phonology* 14 (01): 1–46.

Berko, Jean. 1958. "The Child's Learning of English Morphology." *Word* 14 (2–3): 150–77. https://doi.org/10.1080/00437956.1958.11659661.

Bloomfield, Leonard. 1917. *Tagalog Texts with Grammatical Analysis*. With Alfredo Viola Santiago. University of Illinois.

Boersma, Paul. 1998. "Functional Phonology: Formalizing the Interactions between Articulatory and Perceptual Drives." Ph.D. dissertation, University of Amsterdam.

Boersma, Paul, and Bruce Hayes. 2001. "Empirical Tests of the Gradual Learning Algorithm." *Linguistic Inquiry* 32 (1): 45–86. https://doi.org/10.1162/002438901554586.

Boersma, Paul, and Joe Pater. 2016. "Convergence Properties of a Gradual Learning Algorithm for Harmonic Grammar." In *Harmonic Grammar and Harmonic Serialism*, edited by John McCarthy and Joe Pater. Equinox Publishing Ltd.

Bonami, Olivier, and Gilles Boyé. 2002. "Suppletion and Dependency in Inflectional Morphology." In *Proceedings of the 8th International HPSG Conference*, edited by Frank van Eynde, Lars Hellan, and Dorothee Beermann. CSLI Publications.

Breiss, Canaan, Hironori Katsuda, and Shigeto Kawahara. 2024. "Paradigm Uniformity Is Probabilistic: Evidence from Velar Nasalization in Japanese." *Proceedings of the 39th West Coast Conference on Formal Linguistics* 39.

Broselow, Ellen. 1980. "Syllable Structure in Two Arabic Dialects." *Studies in the Linguistic Sciences* 10 (2): 13–24.

Brysbaert, Marc, and Boris New. 2009. "Moving beyond Kučera and Francis: A Critical Evaluation of Current Word Frequency Norms and the Introduction of a New and Improved Word Frequency Measure for American English." *Behavior Research Methods* 41 (4): 977–90. https://doi.org/10.3758/BRM.41.4.977.

Burnham, Kenneth P., and David R. Anderson. 2002. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. 2nd ed. Springer.

Bynon, Theodora. 1977. *Historical Linguistics*. Cambridge University Press.

Cairo, Alberto. 2013. *The Functional Art: An Introduction to Information Graphics and Visualization*.

Casali, Roderic F. 1997. "Vowel Elision in Hiatus Contexts: Which Vowel Goes?" *Language* 73 (3): 493–533. https://doi.org/10.2307/415882.

Cedergren, Henrietta J., and David Sankoff. 1974. "Variable Rules: Performance as a Statistical Reflection of Competence." *Language* 50 (2): 333–55. https://doi.org/10.2307/412441.

Çetinkaya-Rundel, Mine, and Johanna Hardin. 2024. *Introduction to Modern Statistics*. 2nd ed. OpenIntro.

Chomsky, Noam. 1957. *Syntactic Structures*. Mouton de Gruyter.

Chomsky, Noam. 1965. *Aspects of the Theory of Syntax*. MIT Press.

Chomsky, Noam, and Morris Halle. 1965. "Some Controversial Questions in Phonological Theory." *Journal of Linguistics* 1 (2): 97–214.

Chomsky, Noam, and Morris Halle. 1968. *The Sound Pattern of English*. Harper and Row.

Coetzee, Andries W. 2000. "Post-Nasal Neutralization Phenomena in Tswana; More on *N... Constraints." Unpublished manuscript. University of Massachusetts, Amherst.

Coetzee, Andries W., and Shigeto Kawahara. 2013. "Frequency Biases in Phonological Variation." *Natural Language and Linguistic Theory* 31: 47–89.

Coetzee, Andries W., and Joe Pater. 2008. "Weighted Constraints and Gradient Restrictions on Place Co-Occurrence in Muna and Arabic." *Natural Language & Linguistic Theory* 26 (2): 289–337. https://doi.org/10.1007/s11049-008-9039-z.

Coetzee, Andries W., and Joe Pater. 2011. "The Place of Variation in Phonological Theory." In *The Handbook of Phonological Theory*, 1st ed., edited by John Goldsmith, Jason Riggle, and Alan C. L. Yu, vol. 75. Blackwell.

Cotterell, Ryan, Christo Kirov, John Sylak-Glassman, et al. 2017. "CoNLL-SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection in 52 Languages." Paper presented at CoNLL-SIGMORPHON 2017. *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*. https://doi.org/10.48550/arXiv.1706.09031.

Cowart, Wayne. 1997. *Experimental Syntax: Applying Objective Methods to Sentence Judgements*. Sage Publications.

Culbertson, Jennifer. 2023. "Artificial Language Learning." In *Oxford Handbook of Experimental Syntax*, edited by Jon Sprouse. Oxford University Press.

Daelemans, Walter, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2018. "TiMBL: Tilburg Memory-Based Learner Reference Guide, Version 6.4." https://raw.githubusercontent.com/LanguageMachines/timbl/master/docs/Timbl_6.4_Manual.pdf.

Daland, Robert. 2015. "Long Words in Maximum Entrophy Phontactic Grammars." *Phonology* 32 (3): 353–83. https://doi.org/10.1017/S0952675715000251.

Daland, Robert, Bruce Hayes, James White, Marc Garellek, Andrea Davis, and Ingrid Norrmann. 2011. "Explaining Sonority Projection Effects." *Phonology* 29: 197–234.

Daland, Robert, Mira Oh, and Syejeong Kim. 2015. "When in Doubt, Read the Instructions: Orthographic Effects in Loanword Adaptation." *Lingua* 159: 70–92. https://doi.org/10.1016/j.lingua.2015.03.002.

De Morgan, Augustus. 1847. *Formal Logic, or, the Calculus of Inference, Necessary and Probable*. Taylor and Walton.

Dell, François. 1981. "On the Learnability of Optional Phonological Rules." *Linguistic Inquiry* 12: 31–37.

Della Pietra, Stephen, Vincent Della Pietra, and John Lafferty. 1997. "Inducing Features of Random Fields." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (4): 380–93.

Dixon, R. M. W. 1981. "Wargamay." In *Handbook of Australian Languages*, edited by R. M. W. Dixon and Barry J. Blake, vol. 2. John Benjamins.

Domahs, Ulrike, Ingo Plag, and Rebecca Carroll. 2014. "Word Stress Assignment in German, English and Dutch: Quantity-Sensitivity and Extrametricality Revisited." *The Journal of Comparative Germanic Linguistics* 17 (1): 59–96. https://doi.org/10.1007/s10828-014-9063-9.

Durand, Jacques, Ulrike Gut, and Gjert Kristoffersen, eds. 2014. *The Oxford Handbook of Corpus Phonology*. Oxford University Press.

Eddington, David. 1996. "Diphthongization in Spanish Derviational Morphology: An Empirical Investigation." *Hispanic Linguistics* 8: 1–35.

Eddington, David. 2004. "Issues in Modeling Language Processing Analogically." *Lingua* 114: 849–71.

Eisner, Jason. 1997a. "Efficient Generation in Primitive Optimality Theory." *Proceedings of the 35th Annual Meeting on Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics* (Madrid, Spain), 313–20.

Eisner, Jason. 1997b. "What Constraints Should OT Allow?" Paper presented at 71st Annual meeting of the Linguistic Society of America.

Eisner, Jason. 2000. Review of *Review of Kager: "Optimality Theory,"* by René Kager. *Computational Linguistics* 26 (2): 286–90. https://doi.org/10.1162/coli.2000.26.2.286.

Ellison, T. Mark. 1994. "Phonological Derivation in Optimality Theory." *COLING 1994 Volume 2: The 15th International Conference on Computational Linguistics* 2.

Ernestus, Miriam, and R. Harald Baayen. 2003. "Predicting the Unpredictable: Interpreting Neutralized Segments in Dutch." *Language* 79 (1): 5–38. https://doi.org/10.1353/lan.2003.0076.

Ernestus, Mirjam. 2011. "Gradience and Caregoricality in Phonological Theory." In *The Blackwell Companion to Phonology, Volume IV: Phonological Interfaces*, edited by Marc van Oostendorp, Colin J. Ewen, Elizabeth Hume, and Keren Rice. Wiley-Blackwell. https://doi.org/10.1002/9781444335262.wbctp0089.

Featherston, Sam. 2005. "Magnitude Estimation and What It Can Do for Your Syntax: Some Wh-Constraints in German." *Lingua* 115 (11): 1525–50. https://doi.org/10.1016/j.lingua.2004.07.003.

Field, Andy. 2026. *Discovering Statistics Using R and RStudio*. London. Sage.

Finkel, Raphael, and Gregory Stump. 2007. "Principal Parts and Morphological Typology." *Morphology* 17: 39–75.

Finley, Sarah. 2017. "Locality and Harmony: Perspectives from Artificial Grammar Learning." *Language and Linguistics Compass* 11 (1): e12233.

Flemming, Edward. 2001. "Scalar and Categorical Phenomena in a Unified Model of Phonetics and Phonology." *Phonology* 18 (1): 7–44. https://doi.org/10.1017/s0952675701004006.

Fukazawa, Haruka, and Linda Lombardi. 2003. "Complex Constraints and Linguistic Typology in Optimality Theory." *The Linguistic Review* 20 (2–4): 195–215. https://doi.org/10.1515/tlir.2003.008.

Fuller, Corinna Lee. 2024. "Word-Initial /p/ in Khalkha Mongolian: Variation in Connected Speech." M.A. Thesis, University of California, Los Angeles. https://www.proquest.com/docview/2986554543.

Garcia, Guilherme D. 2017. "Weight Gradience and Stress in Portuguese." *Phonology* 34 (1): 41–79.

Gigerenzer, Gerd. 2001. "The Adaptive Toolbox." In *Bounded Rationality: The Adaptive Toolbox*, edited by Gerd Gigerenzer and R. Selten. MIT Press.

Gigerenzer, Gerd, and Wolfgang Gaissmaier. 2011. "Heuristic Decision Making." *Annual Review of Psychology* 62: 451–82.

Goldwater, Sharon, and Mark Johnson. 2003a. "Learning OT Constraint Rankings Using a Maximum Entropy Model." In *Proceedings of the Stockholm Workshop on Variation within Optimality Theory*, edited by Jennifer Spenader, Anders Eriksson, and Osten Dahl.

Goldwater, Sharon, and Mark Johnson. 2003b. "Learning OT Constraint Rankings Using a Maximum Entropy Model." *Proceedings of the Workshop on Variation within Optimality Theory* (Stockholm University).

Gordon, Matthew. 2002. "A Factorial Typology of Quantity-Insensitive Stress." *Natural Language & Linguistic Theory* 20: 491–552.

Gouskova, Maria, and Suzy Ahn. 2025. "Sublexical Phonotactics and English Comparatives." In *UMOP 42: Papers in Celebration of the 50th Anniversary of UMass Linguistics*, vol. 42. University of Massachusetts Occasional Papers in Linguistics. GLSA, UMASS.

Gouskova, Maria, Newlin-Łukowicz, and Sofya Kasyanenko. 2015. "Selectional Restrictions as Phonotactics over Sublexicons." *Lingua* 167: 41–81.

Greenberg, Joseph H., and James J. Jenkins. 1964. "Studies in the Psychological Correlates of the Sound System of American English." *Word* 20: 157–77.

Guion, Susan G., J. J. Clark, Tetsuo Harada, and Ratree P. Wayland. 2003. "Factors Affecting Stress Placement for English Nonwords Include Syllabic Structure, Lexical Class, and Stress Patterns of Phonologically Similar Words." *Language and Speech* 46 (4): 403–27. https://doi.org/10.1177/00238309030460040301.

Gurevich, Naomi. 2011. "Lenition." In *The Blackwell Companion to Phonology*, 1st ed., edited by Marc van Oostendorp, Colin J. Ewen, Elizabeth Hume, and Keren Rice, vol. 3. Blackwell Publishers.

Guy, Gregory. 2011. "Variability." In *The Blackwell Companion to Phonology*, edited by Marc van Oostendorp, Colin J. Ewen, Elizabeth Hume, and Keren Rice. Wiley-Blackwell.

Guy, Gregory R. 1993. "The Quantitative Analysis of Linguistic Variation." In *American Dialect Research: Celebrating the 100th Anniversary of the American Dialect Society, 1889-1989*, edited by Dennis R. Preston, with John G. Fought. John Benjamins Publishing Company.

Hájek, Alan. 2023. "Interpretations of Probability." In *The Stanford Encyclopedia of Philosophy*, Winter 2023, edited by Edward N. Zalta and Uri Nodelman. https://plato.stanford.edu/archives/win2023/entries/probability-interpret/.

Halle, Morris, and Jean-Roger Vergnaud. 1981. "Harmony Processes." In *Crossing the Boundaries in Linguistics*, edited by W. Klein, J. Hintikka, and S. Peters. Springer.

Hannan, M. 1959. *Standard Shona Dictionary*. St. Martin's Press.

Hansen, Kenneth C., and Hansen, Lesly E. 1969. "Pintupi Phonology." *Oceanic Linguistics* 8 (2): 153–70. https://doi.org/10.2307/3622818.

Hansen, Kenneth C., and Hansen, Lesly E. 1978. *The Core of Pintupi Grammar*. Institute for Aboriginal Development.

Harris, James Wesley. 1969. *Spanish Phonology*. MIT Press.

Hayes, Bruce. 1995. *Metrical Stress Theory: Principles and Case Studies*. University of Chicago Press.

Hayes, Bruce. 2008. *Introductory Phonology*. 1st ed. Wiley-Blackwell.

Hayes, Bruce. 2016. "Comparative Phonotactics." *Proceedings of the 50th Annual Meeting of the Chicago Linguistic Society* 50: 265–85.

Hayes, Bruce. 2017. "Varieties of Noisy Harmonic Grammar." In *Proceedings of the 2016 Annual Meeting on Phonology*, edited by Karen Jesney, Charlie O'Hara, Caitlin Smith, and Rachel Walker, vol. 4. https://doi.org/10.3765/amp.v4i0.3997.

Hayes, Bruce. 2022. "Deriving the Wug-Shaped Curve: A Criterion for Assessing Formal Theories of Linguistic Variation." *Annual Review of Linguistics* 8 (1): 473–94. https://doi.org/10.1146/annurev-linguistics-031220-013128.

Hayes, Bruce, and Zsuzsa Cziráky Londe. 2006. "Stochastic Phonological Knowledge: The Case of Hungarian Vowel Harmony." *Phonology* 23 (1): 59–104. https://doi.org/10.1017/S0952675706000765.

Hayes, Bruce, and Russell G. Schuh. 2019. "Metrical Structure and Sung Rhythm of the Hausa Rajaz." *Language* 95 (2): e253–99. https://doi.org/10.1353/lan.2019.0043.

Hayes, Bruce, and Tanya Stivers. 2000. "Postnasal Voicing." Preprint, University of California, Los Angeles. https://brucehayes.org/papers/HayesAndStivers1995PhoneticsOfPostNasalVoicing.pdf.

Hayes, Bruce, and Colin Wilson. 2008. "A Maximum Entropy Model of Phonotactics and Phontactic Learning." *Linguistic Inquiry* 39 (3): 379–440. https://doi.org/10.1162/ling.2008.39.3.379.

Hayes, Bruce, Kie Zuraw, Péter Siptár, and Zsuzsa Londe. 2009a. "Natural and Unnatural Constraints in Hungarian Vowel Harmony." *Language* 85 (4): 822–63. https://doi.org/10.1353/lan.0.0169.

Hayes, Bruce, Kie Zuraw, Péter Siptár, and Zsuzsa Cziráky Londe. 2009b. "Natural and Unnatural Constraints in Hungarian Vowel Harmony." *Language* 85 (4): 822–63. https://doi.org/10.1353/lan.0.0169.

Hudson Kam, Carla L., and Elissa L. Newport. 2005. "Regularizing Unpredictable Variation: The Roles of Adult and Child Learners in Language Formation and Change." *Language Learning and Development* 1: 151–95.

Hughto, Coral, Andrew Lamont, Brandon Prickett, and Gaja Jarosz. 2019. "Learning Exceptionality and Variation with Lexically Scaled MaxEnt." *Proceedings of the Society for Computation in Linguistics (SCiL)*, 91–101.

Inkelas, Sharon, Aylin Küntav, Orhan Orgun, and Ronald Sprouse. 2000. "Turkish Electronic Living Lexicon (TELL)." *Turkic Languages* 4: 253–75.

Jäger, Gerhard. 2007. "Maximum Entropy Models and Stochastic Optimality Theory." In *Architectures, Rules and Preferences: Variations on Themes by Joan W. Bresnan*, edited by Annie Zaenen, Jane Simpson, Tracy Holloway King, Jane Grimshaw, Joan Maling, and Christopher D. Manning. CSLI Publications.

Jäger, Gerhard, and Anette Rosenbach. 2006. "The Winner Takes It All - Almost: Cumulativity in Grammatical Variation." *Linguistics* 44: 937–71.

Jarosz, Gaja. 2017. "Defying the Stimulus: Acquisition of Complex Onsets in Polish." *Phonology* 34 (2): 269–98.

Jarosz, Gaja, Cerys Hughes, Andrew Lamont, et al. 2025. "Type and Token Frequency Jointly Drive Learning of Morphology." *Journal of Memory and Language* 144: 104666. https://doi.org/10.1016/j.jml.2025.104666.

Jaynes, Edwin T. 2003. *Probability Theory: The Logic of Science*. Edited by G. Larry Bretthorst. Cambridge University Press.

Jo, Jinyoung. 2024a. "Individual Differences in Phonology: The Realization of Stem-Final Coronal Obstruents in Korean." Ph.D. dissertation, UCLA.

Jo, Jinyoung. 2024b. "Korean Vowel Harmony Has Weak Phonotactic Support and Has Limited Productivity." *Phonology* 40 (1–2): 65–100.

Johnson, Mark. 2002. "Optimality-Theoretic Lexical Functional Grammar." In *The Lexical Basis of Setnence Processing: Formal, Computational and Experimental Issues*, edited by Paola Merlo and Suzanne Stevenson, vol. 3. Natural Language Processing 4. John Benjamins Publishing Company.

Jun, Jongho. 2010. "Stem-Final Obstruent Variation in Korean." *Journal of East Asian Linguistics* 19: 137–79.

Jun, Jongho. 2015. "Korean N-Insertion: A Mismatch between Data and Learning." *Phonology* 32 (3): 417–58. https://doi.org/10.1017/S0952675715000275.

Jun, Jongho, and Adam Albright. 2017. "Speakers' Knowledge of Alternations Is Asymmetrical: Evidence from Seoul Korean Verb Paradigms." *Journal of Linguistics* 53 (3): 567–611.

Jurafsky, Daniel, and James H. Martin. 2025. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd ed. https://web.stanford.edu/~jurafsky/slp3/.

Jurafsky, Daniel, and James H. Martin. 2026. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd ed. Online manuscript released January 6. https://web.stanford.edu/~jurafsky/slp3.

Kager, René. 1999. *Optimality Theory*. Cambridge University Press.

Kang, Yoonjung. 2003. "Sound Changes Affecting Noun-Final Coronal Obstruents in Korean." *Japanese/Korean Linguistics* 12: 128–39.

Karttunen, Lauri. 2006. "The Insufficiency of Paper-and-Pencil Linguistics: The Case of Finnish Prosody." In *Intelligent Linguistic Architectures: Variations on Themes by Ronald M. Kaplan*, edited by Miriam Butt, Mary Elizabeth Dalrymple, and Tracy Holloway King. CSLI Publications.

Katsuda, Hironori. 2023. "Loanword Accentuation in Japanese: Corpus Study, Modeling, and Experiments." Ph.D. dissertation, University of California, Los Angeles.

Kaun, Abigail Rhodes. 2004. "The Typology of Rounding Harmony : An Optimality Theoretic Approach." In *Phonetically Based Phonology*, edited by Bruce Hayes, Robert Kirchner, and Donca Steriade. Cambridge University Press.

Kawahara, Shigeto. 2020. "A Wug-Shaped Curve in Sound Symbolism: The Case of Japanese Pokémon Names." *Phonology* 37 (3): 383–418. https://doi.org/10.1017/S0952675720000202.

Kawahara, Shigeto. 2021. "Testing MaxEnt with Sound Symbolism: A Stripy Wug-Shaped Curve in Japanese Pokémon Names." *Language* 97 (4): e341–59. https://doi.org/10.1353/lan.2021.0074.

Kiparsky, Paul. 1973. *Three Dimensions of Linguistic Theory*. Edited by Osamu Fujimura. With Donald L. Smith, S. I. Harada, and Julie B. Lovins. TEC.

Kiparsky, Paul. 1993. "An OT Perspective on Phonological Variation." https://web.stanford.edu/~kiparsky/Papers/nwave94.pdf.

Kiparsky, Paul. 2006a. "A Modular Metrics for Folk Verse." In *Formal Approaches to Poetry: Recent Developments in Metrics*, edited by Bezalel Elan Dresher and Nila Friedberg, vol. 11. Phonology and Phonetics. De Gruyter Mouton.

Kiparsky, Paul. 2006b. "Iambic Inversion in Finnish." In *A Man of Measure: Festschrift in Honour of Fred Karlsson on His 60th Birthday*, edited by Mickael Suominen, Antti Arppe, Anu Airola, et al. Linguistics Association of Finland.

Kirchner, Robert Martin. 2001. *An Effort Based Approach to Consonant Lenition*. 1st ed. Oustanding Dissertations in Linguistics. Routledge.

Kisseberth, Charles W. 1970. "On the Functional Unity of Phonological Rules." *Linguistic Inquiry* 1 (3): 291–306.

Korzybski, Alfred. 1933. *Science and Sanity: An Introduction to Non-Aristotelian Systems and General Semantics*. The International Non-Aristotelian Library Pub. Co.

Kuo, Jennifer. 2023. "Evidence for Prosodic Correspondence in the Vowel Alternations of Tgdaya Seediq." *Phonological Data and Analysis* 5 (3): 1–31. https://doi.org/10.3765/pda.v5art3.77.

Labov, William. 1969. "Contraction, Deletion, and Inherent Variability of the English Copula." *Language* 45 (4): 715–62. https://doi.org/10.2307/412333.

Labov, William. 1973. *Sociolinguistic Patterns*. 1st ed. University of Pennsylvania Press.

Labov, William. 1989. "The Child as Linguistic Historian." *Language Variation and Change* 1: 85–97.

Labov, William. 1994. *Principles of Linguistic Change, Volume 1: Internal Factors*. Blackwell.

Laplace, Pierre-Simon. 1814. "Essai Philosophique Sur Les Probabilités." In *Œuvres Complètes de Laplace*, tome VII. Gauthier-Villars.

Lasdon, Leon S., Richard L. Fox, and Margery W. Ratner. 1974. "Nonlinear Optimization Using the Generalized Reduced Gradient Method." *Revue Française d'automatique, Informatique, Recherche Opérationnelle* 8 (3): 73–104.

Lavoie, Lisa M. 2001. *Consonant Strength: Phonological Patterns and Phonetic Manifestations*. Garland Pub.

Lefkowitz, Lee Michael. 2017. "Maxent Harmonic Grammars and Phonetic Duration." Ph.D. dissertation, Department of Linguistics, UCLA.

Legendre, Géraldine, Yoshiro Miyata, and Paul Smolensky. 1990. "Harmonic Grammar – A Formal Multi-Level Connectionist Theory of Linguistic Well-Formedness: An Application." *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society* (Mahwah, NJ), 884–91.

Linzen, Tal, Sofya Kasyanenko, and Maria Gouskova. 2013. "Lexical and Phonological Variation in Russian Prepositions." *Phonology* 30: 453–515.

Magri, Giorgio. 2025. "Constraint Interaction in Probabilistic Phonology: Deducing MaxEnt from Hayes and Zuraw's Shifted Sigmoids Generalization." *Linguistic Inquiry* 56: 1–54.

Magri, Giorgio, and Edward Flemming. 2024. "Variation." In *The Cambridge Handbook of Phonology*, 2nd ed., edited by Adam Jardine and Paul de Lacy. Cambridge University Press.

Manning, Christopher D. 2003. "Probabilistic Appoaches to Syntax." In *Probabilistic Linguistics*, edited by Rens Bod, Jennifer Hay, and Stefanie Jannedy. MIT Press.

Martin, Andrew. 2011. "Grammars Leak: Modeling How Phonotactic Generalizations Interact within the Grammar." *Language* 87 (4): 751–70.

Mayer, Connor. 2021. "Issues in Uyghur Backness Harmony: Corpus, Experimental, and Computational Studies." Ph.D. dissertation, University of California, Los Angeles.

Mayer, Connor, Adeline Tan, and Kie Zuraw. 2024. "Introducing Maxent.Ot: An R Package for Maximum Entropy Constraint Grammars." *Phonological Data and Analysis* 6 (4): 1–44. https://doi.org/10.3765/pda.v6art4.88.

McCarthy, John J. 1988. "Feature Geometry and Dependency: A Review." *Phonetica* 45 (2): 84–108. https://doi.org/10.1159/000261820.

McCarthy, John J. 2002. *A Thematic Guide to Optimality Theory*. Cambridge University Press.

McCarthy, John J. 2003. "OT Constraints Are Categorical." *Phonology* 20: 75–138.

McCarthy, John J. 2008. *Doing Optimality Theory: Applying Theory to Data*. Blackwell Publishers.

McCarthy, John J., and Alan Prince. 1995. "Faithfulness and Reduplicative Identity." In *Papers in Optimality Theory*, edited by Jill N. Beckman, Laura Walsh Dickey, and Suzanne Urbanczyk, vol. 1. University of Massachusetts Occasional Papers in Linguistics 18. GLSA, UMASS.

McPherson, Laura. 2013. *A Grammar of Tommo So*. De Gruyter.

McPherson, Laura, and Bruce Hayes. 2016. "Relating Application Frequency to Morphological Structure: The Case of Tommo So Vowel Harmony." *Phonology* 33 (1): 125–67. https://doi.org/10.1017/S0952675716000051.

Mielke, Jeff. 2005. "Modeling Distinctive Feature Emergence." In *Proceedings of the West Coast Conference on Formal Linguistics*, edited by John D. Alderete, vol. 24. Cascadilla Proceedings Project.

Moore-Cantwell, Claire. 2016. "The Representation of Probabilistic Phonological Patterns: Neurological, Behavioral, and Computational Evidence from the English Stress System." PhD Thesis, University of Massachusetts Amherst. http://www.clairemoorecantwell.org/dissertation_CMC.pdf.

Moore-Cantwell, Claire. 2020. "Weight and Final Vowels in the English Stress System." *Phonology* 37 (4): 657–95.

Moore-Cantwell, Claire, and Joe Pater. 2016. "Gradient Exceptionality in Maximum Entropy Grammar with Lexically Specific Constraints." *Catalan Journal of Linguistics* 15: 53–66. https://doi.org/10.5565/rev/catjl.183.

Moreton, Elliott, and Shigeaki Amano. 1999. "Phonotactics in the Perception of Japanese Vowel Length: Evidence for Long-Distance Dependencies." *Proceedings of the 6th European Conference on Speech Communication and Technology*, 2679–82.

Moreton, Elliott, Joe Pater, and Katya Pertsova. 2017. "Phonological Concept Learning." *Cognitive Science* 41 (1): 4–69.

Moreton, Elliott, and Katya Pertsova. 2024. "Implicit and Explicit Processes in Phonological Concept Learning." *Phonology* 40 (2): 1–53.

Nakisa, Ramin Charles, Kim Plunkett, and Ulrike Hahn. 2001. "A Cross-Linguistic Comparison of Single and Dual-Route Models of Inflectional Morphology." In *Models of Language Acquisition : Inductive and Deductive Approaches*, edited by Peter Broeder and Jaap Murre. MIT Press.

Odden, David. 1981. "Problems in Tone Assignment in Shona." Ph.D. dissertation, University of Illinois at Urbana-Champaign.

Olejarczuk, Paul, and Vsevolod Kapatsinski. 2018. "The Metrical Parse Is Guided by Gradient Phonotactics." *Phonology* 35 (3): 367–405.

Olson, Erin. 2020. "Loanwords and the Perceptual Map : A Perspective from MaxEnt Learning." Ph.D. dissertation, Massachusetts Institute of Technology.

Orr, Carolyn. 1962. "Ecuadorian Quichua Phonology." In *Studies in Ecuadorian Indian Languages I*, 1st ed., edited by Benjamin F. Elson, with Catherine Peeke. Linguistic Series 7. Summer Institute of Linguistics of the University of Oklahoma.

Pater, Joe. 2000. "Non-Uniformity in English Secondary Stress: The Role of Ranked and Lexically Specific Constraints." *Phonology* 17 (2): 237–74. https://doi.org/10.1017/S0952675700003900.

Pater, Joe. 2004. "Austronesian Nasal Substitution and Other NC Effects." In *Optimality Theory in Phonology: A Reader*, 1st ed., edited by John J. McCarthy. Blackwell.

Pater, Joe. 2009. "Weighted Constraints in Generative Linguistics." *Cognitive Science* 33 (6): 999–1035. https://doi.org/10.1111/j.1551-6709.2009.01047.x.

Pater, Joe. 2010. "Morpheme-Specific Phonology: Constraint Indexation and Inconsistency Resolution." In *Phonological Argumentation: Essays on Evidence and Motivation*, edited by Steve Parker. Advances in Optimality Theory, edited by Ellen Woolford and Armin Mester. University of Toronto Press.

Pater, Joe. 2016. "Universal Grammar with Weighted Constraints." In *Harmonic Grammar and Harmonic Serialism*, edited by John McCarthy and Joe Pater. Equinox.

Pierrehumbert, Janet. 2006. "The Statistical Basis of an Unnatural Alternation." In *Laboratory Phonology 8: Varieties of Phonological Competence*, edited by Louis Goldstein, D. H. Whalen, and Catherine Best. Berlin: Mouton de Gruyter.

Pierrehumbert, Janet. 2022. "More than Seventy Years of Probabilistic Phonology." In *The Oxford History of Phonology*, edited by B. Elan Dresher and Harry van der Hulst. Oxford University Press. https://doi.org/10.1093/oso/9780198796800.003.0030.

Pierrehumbert, Janet B. 1993. "Dissimilarity in the Arabic Verbal Roots." *Proceedings of the North East Linguistics Society 23* (University of Ottawa) 2: 367–81.

Piñeiro, Gervasio, Susana Perelman, Juan P. Guerschman, and José M. Paruelo. 2008. "How to Evaluate Models: Observed vs. Predicted or Predicted vs. Observed?" *Ecological Modeling* 216: 316–22.

Podesva, Robert, and Devyani Sharma, eds. 2013. *Research Methods in Linguistics*. Cambridge University Press.

Poplack, Shana, Alicia Pousada, and David Sankoff. 1982. "Competing Influences on Gender Assignment: Variable Process, Stable Outcome." *Lingua* 57 (1): 1–28. https://doi.org/10.1016/0024-3841(82)90068-7.

Potts, Christopher, Joe Pater, Karen Jesney, Rajesh Bhatt, and Michael Becker. 2010. "Harmonic Grammar with Linear Programming: From Linear Systems to Linguistic Typology." *Phonology* 27 (1): 77–117.

Prentice, D. J. 1971. *The Murut Languages of Sabah*. Australian National University.

Prince, Alan. 2003a. "Anything Goes." In *A New Century of Phonology and Phonological Theory : A Festschrift Professor Shosuke Haraguchi on the Occasion of His Sixtieth Birthday*, edited by T. Honma, M. Okazaki, T. Tabata, and S. Tanaka. Kaitakusha.

Prince, Alan. 2003b. "Arguing Optimality." *University of Massachusetts Occasional Papers in Linguistics* 26: 268–304.

Prince, Alan, and Paul Smolensky. 1993. *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell.

Prince, Alan, and Paul Smolensky. 2004. *Optimality Theory : Constraint Interaction in Generative Grammar*. Blackwell Publishers.

Riggle, Jason. 2009. "Generating Contenders." Preprint, Rutgers Optimality Archive. https://roa.rutgers.edu/files/1044-0809/1044-RIGGLE-0-0.PDF.

Riggle, Jason. 2010. "Sampling Rankings." Preprint, Rutgers Optimality Archive.

Ringen, Catherine O., and Orvokki Heinämäki. 1999. "Variation in Finnish Vowel Harmony: An OT Account." *Natural Language & Linguistic Theory* 17 (2): 303–37. https://doi.org/10.1023/a:1006158818498.

Rosenthall, Samuel. 1989. "The Phonology of Nasal-Obstruent Sequences." M.A. Thesis, Department of Linguistics, McGill University. https://www.proquest.com/dissertations-theses/phonology-nasal-obstruent-sequences/docview/303802768/se-2?accountid=14512.

Rousseau, Pascale, and David Sankoff. 1978. "Advances in Variable Rule Methodology." In *Linguistic Variation : Models and Methods*. Academic Press.

Sankoff, David. 1988. "Variable Rules." In *Sociolinguistics : An International Handbook of the Science of Language and Society*, 2nd ed., edited by Ulrich Ammon, Norbert Dittmar, and Klaus J. Mattheier, vol. 2. De Gruyter.

Sankoff, David, and William Labov. 1979. "On the Uses of Variable Rules." *Language in Society* 8 (2–3): 189–222. https://doi.org/10.1017/S0047404500007430.

Schachter, Paul, and Fe T. Otanes. 1972. *Tagalog Reference Grammar*. University of California Press.

Scholes, Robert J. 1966. *Phonotactic Grammaticality*. Mouton & Co.

Sereno, Joan A., and Allard Jongman. 1997. "Processing of English Inflectional Morphology." *Memory and Cognition* 25 (4): 425–37.

Shaw, Jason, and Shigeto Kawahara. 2018. "Predictability and Phonology: Past, Present and Future." Article No. 20180042. *Linguistic Vanguard* 4 (2): 1–11. https://doi.org/10.1515/lingvan-2018-0042.

Shih, Stephanie S. 2017. "Constraint Conjunction in Probabilistic Weighted Constraint Grammar." *Phonology* 34 (2): 243–68. https://doi.org/10.1017/S0952675717000136.

Shih, Stephanie S. 2018. "Learning Lexical Classes from Variable Phonology." *ICUWPL* 4: 1–15.

Simpson, Edward H. 1951. "The Interpretation of Interaction in Contigency Tables." *Journal of the Royal Statistical Society, Series B* 13: 238–41.

Skousen, Royal, Deryle Lonsdale, and Dilworth P. Parkinson, eds. 2002. *Analogical Modeling: An Exemplar-Based Approach to Language*. John Benjamins Publishing Company.

Smith, Brian W., and Joe Pater. 2020. "French Schwa and Gradient Cumulativity." *Glossa: A Journal of General Linguistics* 5 (1): 24–33. https://doi.org/10.5334/gjgl.583.

Smolensky, Paul. 1986. "Information Processing in Dynamical Systems: Foundations of Harmony Theory." In *Parallel Distributed Processing : Explorations in the Microstructure of Cognition*, by David E. Rumelhart, James L. McClelland, and the PDP Research Group, vol. 1. MIT Press.

Smolensky, Paul, and Géraldine Legendre. 2006. *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar*. MIT Press.

Solé, Maria-Josep, Patrice Speeter Beddor, and Manjari Ohala, eds. 2007. *Experimental Approaches to Phonology*. Oxford University Press.

Storme, Benjamin. 2026. "A Method to Evaluate Systemic Constraints in Probabilistic Grammars." *Linguistic Inquiry* 57 (1): 216–28.

Suomi, Kari. 1983. "Palatal Vowel Harmony: A Perceptually Motivated Phenomenon?" *Nordic Journal of Linguistics* 6 (1): 1–35. https://doi.org/10.1017/S0332586500000949.

Szmrecsanyi, Benedikt, Jason Grafmiller, Joan Bresnan, et al. 2017. "Spoken Syntax in a Comparative Perspective: The Dative and Genitive Alternation in Varieties of English." *Glossa* 2 (1): 1–27. https://doi.org/10.5334/gjgl.310.

Tesar, Bruce. 2004. "Computing Optimal Forms in Optimality Theory: Basic Syllabification." In *Optimality Theory in Phonology: A Reader*, edited by John J. McCarthy. Blackwell Publishing Ltd. https://doi.org/10.7282/T398852N.

Tesar, Bruce, and Paul Smolensky. 2000. *Learnability in Optimality Theory*. MIT Press.

Tufte, Edward. 2001. *The Visual Display of Quantitative Information*. 2nd ed. Graphics Press.

Tukey, John W. 1980. "We Need Both Exploratory and Confirmatory." *The American Statistician* 34.

Vendelin, Inga, and Sharon Peperkamp. 2006. "The Influence of Orthography on Loanword Adaptations." *Lingua* 116 (7): 996–1007. https://doi.org/10.1016/j.lingua.2005.07.005.

Walker, James A. 2010. *Variation in Linguistic Systems*. 1st ed. Routledge.

Wasserman, Larry. 2004. *All of Statistics: A Concise Course in Statistical Inference*. Springer.

Westbury, John R., and Patricia A. Keating. 1986. "On the Naturalness of Stop Consonant Voicing." *Journal of Linguistics* 22 (1): 145–66. https://doi.org/10.1017/S0022226700010598.

White, Lydia. 2003. *Second Language Acquisition and Universal Grammar*. Cambridge Textbooks in Linguistics. Cambridge University Press.

Wilson, Colin. 2006. "Learning Phonology with Substantive Bias: An Experimental and Computational Study of Velar Palatalization." *Cognitive Science* 30 (5): 945–82. https://doi.org/10.1207/s15516709cog0000_89.

Wilson, Colin, and Gillian Gallagher. 2018. "Accidental Gaps and Surface-Based Phonotactic Learning: A Case Study of South Bolivian Quechua." *Linguistic Inquiry* 49 (3): 610–23.

Wilson, Colin, and Marieke Obdeyn. 2009. "Simplifying Subsidiary Theory: Statistical Evidence from Arabic, Muna, Shona and Wargamay." M.S. Thesis, Johns Hopkins University.

Wolfram, Walt. 1993. "Identifying and Interpreting Variables." In *American Dialect Research*, edited by Dennis R. Preston. John Benjamins Publishing Company.

Xu, Lily. 2025. "Non-Concatenative Morphological Domains Constraint Phonotactics: A Case Study of Egyptian Arabic." *Phonology* 42: e16.

Yang, Charles. 2016. *The Price of Linguistic Productivity: How Children Learn to Break the Rules of Language*. MIT Press.

Zhang, Jie, Yuwen Lai, and Craig Turnbull-Sailor. 2006. "Wug-Testing the 'Tone Circle' in Taiwanese." In *Proceedings of the 25th West Coast Conference on Formal Linguistics*, edited by Donald Baumer, David Montero, and Michael Scanlon. Cascadilla Press.

Zhang, Jie, Yuwen Lai, and Craig Turnbull-Sailor. 2009. "Opacity, Phonetics, and Frequency in Taiwanese Tone Sandhi." In *Current Issues in Unity and Diversity of Languages: Collection of Papers Selected from the 18th International Congress of Linguists*. Linguistic Society of Korea.

Zimmer, Karl E. 1969. "Psychological Correlates of Some Turkish Morpheme Structure Conditions." *Language* 45 (2): 309–21. https://doi.org/10.2307/411662.

Zuraw, Kie. 2002. "Aggressive Reduplication." *Phonology* 19 (3): 395–439. https://doi.org/10.1017/S095267570300441X.

Zuraw, Kie. 2010. "A Model of Lexical Variation and the Grammar with Application to Tagalog Nasal Substitution." *Natural Language & Linguistic Theory* 28 (2): 417–72. https://doi.org/10.1007/s11049-010-9095-z.

Zuraw, Kie, and Bruce Hayes. 2017. "Intersecting Constraint Families: An Argument for Harmonic Grammar." *Language* 93 (3): 497–548.

Zuraw, Kie, Isabelle Lin, Meng Yang, and Sharon Peperkamp. 2021. "Competition between Whole-Word and Decomposed Representations of English Prefixed Words." *Morphology* 31: 201–37.

Zuraw, Kie Ross. 2000. "Patterned Exceptions in Phonology." University of California, Los Angeles.

Zymet, Jesse. 2018. "Learning a Frequency-Matching Grammar Together with Lexical Isiosyncrasy: MaxEnt vs. Heirarchical Regression." *Proceedings of the Annual Meeting on Phonology*, ahead of print. https://doi.org/10.3765/amp.v7i0.4495.