

EXTENDING HIDDEN STRUCTURE LEARNING:
FEATURES, OPACITY, AND EXCEPTIONS

A Dissertation Presented

By

ALEKSEI IOULEVITCH NAZAROV

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2016

Linguistics

© Copyright by Aleksei Ioulevitch Nazarov 2016
All Rights Reserved

EXTENDING HIDDEN STRUCTURE LEARNING:
FEATURES, OPACITY, AND EXCEPTIONS

A Dissertation Presented

By

ALEKSEI IOULEVITCH NAZAROV

Approved as to style and content by

Joe Pater, Co-Chair

Gaja Jarosz, Co-Chair

John McCarthy, Member

Kristine Yu, Member

David Smith, Member

John Kingston, Department Head
Department of Linguistics

DEDICATION

For my parents, Elena and Yuli.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my dissertation co-chairs, Gaja Jarosz and Joe Pater, for their patience, guidance, and support in the process of writing this dissertation.

I am very grateful to Gaja for being willing to take the responsibility of becoming my co-chair in her first year of joining the UMass faculty, and my last year as a UMass graduate student. I would like to thank her for her enormous encouragement as well as her constructive criticism, and for opening my mind onto new perspectives and teaching me how to better articulate my thoughts. Without the many things that I learned from her in the course of this one year, this dissertation would not have taken the course it has taken now.

I am especially thankful to Joe for helping me realize that I wanted to do computational phonology by encouraging me to explore this direction since my first year of graduate school. I am also very thankful to him as a teacher: the thoughts and ideas that came out of his courses will always stay with me; he has taught me to always be curious for the road not taken. His flexibility and guidance in the process of shaping this dissertation have been invaluable. Without exaggeration, I can say that without this, this dissertation would not exist at all.

I would also like to thank John McCarthy for his patience, wisdom, and sense of humor. Working with him throughout my time as a graduate student has taught me to be infinitely more critical of my own thinking and writing. I will never forget the anecdotes and pieces of wisdom that he shared during meetings and classes, and I thank him for being generous in everything.

Kristine Yu deserves my special thanks for her encouragement and guidance, especially in the project that led to chapter 2 of this dissertation. I am very thankful to her for teaching me to be more clear and precise, and for broadening my horizons with literature and perspectives I was completely unaware of before.

I would like to thank David Smith for being very supportive and flexible, and for his very valuable comments on various instantiations of the projects explored in this dissertation. I learned many things about matters both computational and linguistic – things that will also help me to keep growing in the future.

Brian Dillon and John Kingston have also been very important in the process of developing this dissertation, and I am very grateful for their help and support. Brian Dillon co-advised the project that led to chapter 2 of this dissertation. I would like to thank him for guiding me through my first steps in developing a computational learning project, for his support, for his inspiring ideas, and for all his help with both technical and linguistic questions in this project. In the early stages of exploring dissertation themes, I considered an experimental phonology project, with John Kingston as the prospective chair of the committee. I would like to thank John for his commitment and support, and for his help with experiment design and recording stimuli. I am very grateful for all that I learned from him during this process.

I am very grateful to all of the UMass faculty for these 6 years of growth and learning. Apart from those whose name I have already mentioned, I would also like to especially thank Rajesh Bhatt, Lisa Green, Alice Harris, Kyle Johnson, Angelika Kratzer, Tom Roeper, and Ellen Woolford, who I have learned incredibly much from either in class or in person. I would also like to thank David Huber, Alexandra Jesse, Andrew

McCallum, and Erik Learned-Miller for their courses that exponentiated my knowledge of modeling from various angles and perspectives, and that gave me the tools that make me not ashamed to call myself a computational phonologist.

I would like to thank Marc van Oostendorp for being my first advisor at Leiden University. He has always supported and encouraged me to continue my path in linguistics, and I would like to thank him for being a true inspiration, both as a linguist and as a person. I would also like to thank all other linguistics faculty at Leiden University for all that I have learned from them and for their support, in particular, Lisa Cheng, Crit Cremers, and Claartje Levelt.

I would like to thank all the wonderful colleagues who have commented on various parts of this work throughout the years, in particular, Adam Albright, Ricardo Bermúdez-Otero, Paul Boersma, Jeroen Breteler, Lisa Davidson, Brian Dillon, David Erschler, Silke Hamann, Ivy Hauser, Coral Hughto, Martin Krämer, John Kingston, John McCarthy, Marc van Oostendorp, Joe Pater, Presley Pizzo, Clàudia Pons-Moll, Olivier Rizzolo, Klaas Seinhorst, Robert Staubs, Peter Svenonius, and Kristine Yu, and audiences at UMass Amherst, the University of Tromsø, Leiden University, the University of Manchester, the LSA Annual Meeting in Washington.

I would also like to thank the National Science Foundation for supporting the work that led to chapter 3 of this dissertation through grants BCS-0813829 and BCS-1424007.

For rescuing me time and time again, I am deeply indebted to Kathy Adamczyk, Sarah Vega-Liros, Tom Maxfield, and Michelle McBride. Thank you for keeping our

department running, thank you for your company and moral support, and thank you for your incredible flexibility and resourcefulness.

The UMass linguistics department has been a place of contemplation, ideas, and wonderful connections with people. I would like to thank all my amazing fellow graduate students, and I would especially like to thank my cohort – Tracy Conner, Nick LaCara, and Yangsook Park. I would also like to thank the community of phonologists at UMass for the stimulating discussions and fun get-togethers, especially Ivy Hauser, Coral Hughto, Claire Moore-Cantwell, Presley Pizzo, Brandon Prickett, Brian Smith, and Robert Staubs.

I would like to thank David Erschler for being a true friend, and for putting up with my rants about cooking shows and pulp culture. I am also really thankful to Claire Moore-Cantwell for our conversations about life, music and phonology. I would like to thank Rodica Ivan, Claire Moore-Cantwell, Nick LaCara, Robert Staubs, and Katya Vostrikova for being amazing roommates and friends throughout the years. A big thank you also to Caroline Andrews, Thuy Bui, Seda Kan, Stefan Keine, Deniz Özyıldız, and Brian Smith for being kind, supportive, and otherwise wonderful human beings.

I am also very grateful for the support I have gotten from my colleagues outside of this department. In particular, I would like to thank Michael Becker, Gašper Beguš, Clàudia Pons-Moll, Kevin Schluter, Xico Torres-Tamarit, Asia Zaleska for being truly supportive and kind, and generally amazing. I would like to thank Mirjam de Jonge not only for being an amazing friend in general, but also for being my longest-standing friend in linguistics – we met during a linguistics course that we took when we were still in secondary school, and we're both still at it.

I am especially indebted to all the friends that I was fortunate to make in this area outside of linguistics, in particular to Anthony, Bobby, Charlie, Ilya, Julian, Jordan, Megan, and Zarina. I would not have been able to do this without your support, and you know I mean it. Thank you for all you have taught me throughout these years. All my friends from elsewhere, you know who you are, thank you for always being there for me; and thank you especially to Bernard, Hannah, Ilia, Inas, Jacco, Jacob, James, Joram, Karima, Lai, Maarten, Nate, and Qimin.

I am immensely grateful to Mallorie Chernin and the Amherst College Choral Society for teaching me so much as a human being and as a singer, and to Mark Swanson for giving me the opportunity to be a soloist in his Rodgers and Hammerstein project. I am also very thankful to the Amherst College Russian Club, and Maria, Nina, and Evgenia for organizing our weekly tea with music and singing. It has truly been an amazing time.

My parents deserve much more gratitude than this page can contain. To my father, Yuli, I am incredibly thankful for teaching me how to think, to be critical, and to persevere; for always supporting me in my decision to go to the US, and for giving me advice from academic to academic. To my mother, Elena, I am immensely grateful for first introducing me to linguistics; for always supporting me despite my flaws; and for instilling in me a sense of responsibility, wonder and, most importantly, humor.

Finally, I would like to thank the rest of my family - my brothers and my grandmothers -, and all of the community back home for always being there for me and supporting me, and I would especially like to thank Fr. Meletios for his unwavering support, guidance, and advice.

ABSTRACT

EXTENDING HIDDEN STRUCTURE LEARNING: FEATURES, OPACITY, AND EXCEPTIONS

SEPTEMBER 2016

ALEKSEI IOULEVITCH NAZAROV

B.A., LEIDEN UNIVERSITY

M.PHIL., LEIDEN UNIVERSITY

PH.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Gaja Jarosz and Professor Joe Pater

This dissertation explores new perspectives in phonological hidden structure learning (inferring structure not present in the speech signal that is necessary for phonological analysis; Tesar 1998, Jarosz 2013a, Boersma and Pater 2016), and extends this type of learning towards the domain of phonological features, towards derivations in Stratal OT (Bermúdez-Otero 1999), and towards exceptionality indices in probabilistic OT. Two more specific themes also come out: the possibility of inducing instead of pre-specifying the space of possible hidden structures, and the importance of cues in the data for triggering the use of hidden structure. In chapters 2 and 4, phonological features and exception groupings are induced by an unsupervised procedure that finds units not explicitly given to the learner. In chapters 2 and 3, there is an effect of non-specification or underspecification on the hidden level whenever the data does not give enough cues for that hidden level to be used. When features are hidden structure (chapter 2), they are only used for patterns that generalize across multiple segments. When intermediate derivational levels are hidden structure (chapter 3), the hidden structure necessary for opaque interactions is found more often when additional cues for the stratal affiliation of the opaque process are present in the data.

Chapter 1 motivates and explains the central questions in this dissertation. Chapter 2 shows that phonological features can be induced from groupings of segments (which is motivated by phonetic non-transparency of feature assignment, see, e.g., Anderson 1981), and that patterns that do not generalize across segments are formulated in terms of segments in such a model. Chapter 3 implements a version of Stratal OT (Bermúdez-Otero 1999), and confirms Kiparsky's (2000) hypothesis that evidence for an opaque process' stratal affiliation makes it easier to learn an opaque interaction, even when opaque interactions are more difficult to learn than their transparent counterparts. Chapter 4 proposes a probabilistic (instead of non-probabilistic; e.g. Pater 2010) learner for lexically indexed constraints (Pater 2000) in Expectation Driven Learning (Jarosz submitted), and demonstrates its effectiveness on Dutch stress (van der Hulst 1984, Kager 1989, Nouveau 1994, van Oostendorp 1997).

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS.....	v
ABSTRACT.....	x
LIST OF TABLES.....	xvi
LIST OF FIGURES.....	xix
 CHAPTER	
1. INTRODUCTION.....	1
2. THE EMERGENCE OF REDUNDANCY BETWEEN FEATURES AND SEGMENTS.....	12
2.1 Introduction.....	12
2.1.1 Emergentist models of linguistic structure.....	14
2.1.2 Features as a way to increase generality.....	16
2.1.2.1 The necessity and use of features in the grammar....	16
2.1.2.2 Motivating the presence of features through generality.....	17
2.1.2.3 Implementing generality.....	18
2.1.3 Why features are hidden structure and how to learn them....	21
2.1.3.1 Classificatory phonological features: not always phonetically transparent.....	21
2.1.3.2 Learning classificatory features from phonological patterning.....	24
2.1.4 Learning grammars: phonotactic constraint induction and weighting.....	27
2.1.5 Testing the model and results prediction.....	29
2.1.6 Chapter preview.....	32
2.2. A radically emergentist model of grammar and feature learning.....	32
2.2.1 Classificatory phonological features as byproducts of grammar learning.....	32
2.2.2 The components of the model.....	39
2.2.3 Data.....	42
2.2.4 Intuitions about predictions.....	48
2.3 Computational implementation.....	53
2.3.1 General structure of the algorithm.....	53
2.3.2 The steps of the algorithm.....	56

2.3.2.1 Selection of constraints.....	56
2.3.2.1.1 Maximum Entropy learning for phonotactics.....	57
2.3.2.1.2 Information gain.....	61
2.3.2.1.3 Constraint selection.....	62
2.3.2.2 Context creation.....	65
2.3.2.3 Clustering.....	66
2.3.2.4 Feature induction.....	68
2.3.2.5 Constraint weighting.....	74
2.4 Results.....	78
2.4.1 No word-final [m].....	81
2.4.2 No word-initial nasals.....	82
2.4.3 No labials between high vowels.....	83
2.4.4 Summary of results.....	85
2.5 Discussion and conclusion.....	86
3. LEARNING OPACITY IN STRATAL MAXENT: OPACITY AND EVIDENCE FOR STRATAL AFFILIATION.....	95
3.1 Introduction.....	95
3.2 Case study I: Southern French tensing/laxing.....	100
3.2.1 Data and analysis.....	100
3.2.2 Learning.....	106
3.2.3 Results.....	109
3.3 Case study II: Raising and flapping in Canadian English.....	110
3.3.1 Data.....	110
3.3.2 Simulation setup.....	112
3.3.3 Results.....	117
3.3.4 Summary.....	120
3.4. Difficulties in learning hidden structure.....	120
3.4.1 Cross-level dependencies.....	120
3.4.2 Advantage from evidence for stratal affiliation.....	125
3.5. Concluding remarks.....	127
4. LEARNING THE DIVIDE BETWEEN RULE AND EXCEPTIONS FOR DUTCH STRESS.....	131
4.1 Introduction.....	131
4.1.1 Lexically Indexed Constraint Theory.....	136
4.1.2 Other approaches to exceptionality.....	141
4.1.3 Paper overview.....	142

4.2 Dutch primary stress assignment.....	142
4.2.1 The generalizations.....	142
4.2.1.1 Non-exceptional stress.....	142
4.2.1.2 Exceptions.....	144
4.2.1.3 Psycholinguistic evidence.....	146
4.2.2 OT analysis of Dutch main stress.....	153
4.2.2.1 Non-exceptional stress.....	154
4.2.2.1.1 Implementation.....	155
4.2.2.2 Exceptional stress.....	161
4.2.2.2.1 Final stress.....	162
4.2.2.2.2 Exceptional penultimate stress.....	165
4.2.2.2.3 Exceptional antepenultimate stress.....	168
4.2.2.2.4 Summary of analysis.....	171
4.3 The learning framework: inducing lexically indexed constraints.....	174
4.3.1 Expectation Driven Learning.....	175
4.3.1.1 Pairwise ranking probabilities.....	176
4.3.1.2 The generator.....	177
4.3.1.3 Expectation Maximization.....	178
4.3.2 Inducing indexed constraints.....	183
4.3.3 Phonotactic learning.....	189
4.3.4 Summary of learner.....	191
4.4 Simulations and results.....	193
4.4.1 Training data.....	193
4.4.2 Simulation setup.....	196
4.4.3 Simulation results.....	197
4.4.3.1 Accuracy on training data.....	198
4.4.3.2 Testing on novel items.....	199
4.4.3.3 Evaluating rankings.....	200
4.4.3.3.1 Erroneous antepenultimate stress in learner without phonotactic stage.....	203
4.4.3.3.2 Erroneous antepenultimate stress in XHH in learner with phonotactic stage.....	204
4.4.3.3.3 Full ranking probability tables for both learners.....	207
4.4.3.4 Accounting for exceptions: indexed constraints.....	209
4.4.3.4.1 Indexed constraints found for a low- performance run.....	210
4.4.3.4.2 Indexed constraints found for a high- performance run.....	215
4.4.3.5 Summary of results.....	220
4.5 Conclusion.....	221

APPENDIX A. ONE-SEGMENT PHONOLOGICAL PATTERNS AS FOUND IN MIELKE (2007).....	226
APPENDIX B. FULL RESULTS FOR FEATURE LEARNING SIMULATIONS DESCRIBED IN CHAPTER 2.....	229
APPENDIX C. TRAINING DATA FOR THE SIMULATIONS DESCRIBED IN CHAPTER 4.....	233
APPENDIX D. ALL METRICAL PARSES CONSIDERED IN THE SIMULATIONS DESCRIBED IN CHAPTER 4.....	234
BIBLIOGRAPHY.....	235

LIST OF TABLES

Table	Page
1. Segment inventory of the toy language.....	47
2. Context-and-filler table for $\{*\#m, *\#n, *\#\eta\}$: all possible contexts are on the vertical dimension, all possible fillers are on the horizontal dimension ...	66
3. Context-and-filler table values for $*\#_$	67
4. Division into higher-mean (bolded) and lower-mean component (not bolded).....	68
5. Likelihood vector for higher mean Gaussian in context $*\#_$	69
6. A likelihood vector non-identical to the one in Table 5, similarity < 0.9	72
7. A likelihood vector non-identical to the one in Table 5, similarity > 0.9	73
8. Grammar with constraints $*\#m, *\#n, *\#\eta$	75
9. Reset of specific constraints to zero.....	76
10. Results of example run.....	79
11. Non-zero constraints from Table 10.....	80
12. Grammars that appeal to strange “features”.....	83
13. Other constraints for “no labials between high vowels”.....	85
14. Type of unit appealed to for each phonotactic pattern.....	85
15. Expected probabilities of UR/SR pairings: sum over all derivational paths.....	106
16. Results for the Southern French data set.....	109
17. UR-to-SR probabilities for the local optimum.....	110
18. Weights that generate the local optimum.....	110
19. Datasets for Canadian English.....	112
20. Sample successful weights for opaque ‘mitre-cider-life-lie-for’	114

21. Surface form probabilities generated by the graphs in Figure 9.....	116
22. Sample weights for successful runs of various transparent datasets.....	117
23. Results for Canadian English, for 100 runs.....	117
24. Local optima found for Canadian English simulations.....	118
25. Sample weights for local optima.....	119
26. Sample initialization that leads to local optimum for ‘mitre-cider’.....	121
27. Observed distribution for opaque ‘mitre’ and ‘cider’.....	122
28. KL-divergence for grammars with varying weights of phrase level Ident(low) and all other weights as in Table 26.....	123
29. KL-divergence for ‘mitre-cider’ when phrase level Ident(low) has zero weight.....	124
30. Adding ‘life’ leads to a stronger effect of representing raising at the word level.....	126
31. Adding ‘lie for’ makes it less attractive to lower weight on Ident(low).....	127
32. Number of Dutch monomorphemic words for the relevant stress-and- syllable-weight combinations, according to Ernestus and Neijt (2008); numbers shown in gray are considered to be negligibly small (smaller than 5).....	145
33. Adult judgments from Nouveau (1994): percentage of antepenultimate, penultimate, and final stress among nonce word pronunciations, per weight type.....	149
34. Success rates of imitating main stress patterns in nonce words per weight pattern for 3 and 4 year old children. Derived from error rate tables (48) in Nouveau (1994:130) and (50) <i>ibid</i> :133.....	150
35. Adult judgments Ernestus and Neijt (2008): percentage of antepenultimate, penultimate, and final stress among nonce word pronunciations, per weight type.....	152
36. Example of probabilistic grammar for EDL: Categorical grammar A >> C >> B.....	177

37. Example of probabilistic grammar for EDL: Categorical ranking A, C >> B; ranking of A and C probabilistic.....	177
38. Example of probabilistic grammar for EDL: All rankings are probabilistic.....	177
39. Frequencies of words in learning data per weight and stress type (based on real Dutch frequencies in Table 32).....	194
40. Results from testing grammars obtained without phonotactic learning.....	199
41. Results from testing grammars obtained with 40 iterations of phonotactic learning.....	199
42. Typical ranking for the first run of the simulation without a phonotactic learning stage.....	201
43. Typical ranking for the first run of the simulation with a phonotactic learning stage.....	201
44. Ranking probabilities table for the first run of the simulation without a phonotactic learning stage.....	208
45. Ranking probabilities table for the first run of the simulation with a phonotactic learning stage.....	209
46. Words' partiality to indexed constraints per weight and stress type; low- performance run.....	212
47. Ranking probabilities for all indexed constraints in sample analysis found for low-performance run.....	214
48. Words' partiality to indexed constraints per weight and stress type; high- performance run.....	216
49. Ranking probabilities for all indexed constraints in sample analysis found for high-performance run.....	219
50. List of all words given to the learner with their main stress patterns.....	233
51. List of all parses considered by the learner.....	234

LIST OF FIGURES

Figure	Page
1. The relation between phonetic and classificatory features.....	34
2. Innate features model.....	36
3. Emergent categories model.....	37
4. Cyclic model of grammar learning.....	40
5. Summary of the model	42
6. Different levels of abstraction at which a sound transcribed as [m] can be represented.....	88
7. Word level derivation graphs for /set#a/ and /se#ta/.....	104
8. Graphs for the complete derivation of “this A” and “it is ‘ta’”.....	105
9. Derivation graphs for opaque Canadian Raising.....	114
10. Derivation graph (‘mitre’ only) for weights in Table 26.....	122
11. Hasse diagram for the first run of the simulation without a phonotactic learning stage.....	202
12. Hasse diagram for the first run of the simulation with a phonotactic learning stage.....	202
13. Hasse diagram for analysis found for high-performance run.....	218

CHAPTER 1

INTRODUCTION

The problem of learning a phonological grammar involves hidden structure (Johnson 1984, 1992, Elman and Zipser 1988, Dresher and Kaye 1990, Tesar 1998, 2004, 2006, 2008, 2011, Tesar et al. 2003, Prince and Tesar 2004, Alderete et al. 2005, Jarosz 2006a, 2006b, Jarosz 2013a, 2013b, Merchant 2008, Merchant and Tesar 2008, Akers 2011): structure absent in the learning data that must be inferred by the learner. In this dissertation, I propose several novel methods of learning with hidden structure that operate in various domains that have been understudied in most of the phonological learning literature: phonological features, derivational ordering, and exceptionality.

To my knowledge, I present the first formal phonological learner in which phonological features are treated as hidden structure: they have only been treated as hidden structure in general-purpose learners that do not claim to learn a grammar (Lin 2005, Lin and Mielke 2008). Derivational ordering (Jarosz 2015, Staubs and Pater 2016) and exceptionality (Becker 2009, Coetzee 2009b, Pater 2010) have been treated as hidden structure in phonological grammar learning before, but I propose (in co-authorship with Joe Pater) the first serial learner for a version of Stratal Optimality Theory (Bermúdez-Otero 1999, Kiparsky 2000), and the first learner that induces lexically specific constraints (Pater 2000) in a probabilistic learning framework (see, for instance, Jarosz 2013a for the benefits of a probabilistic framework for learning hidden structure in general, and see Coetzee and Pater 2011 for the benefits of a probabilistic framework for accounting for variation).

In this way, this dissertation expands current horizons on how to realistically

approach a phonological learning problem, as well as offering new tools to deal with these new aspects of the problem. Apart from the specific contributions of each separate project, there are also two broader points that run across the chapters of the dissertation. The first point is that hidden structure can be inferred even when the units out of which the structure should be built are not known. This is the case in chapter 2, where the grammar only receives a procedure for finding features, but not the features themselves. This is also the case in chapter 4, where the grammar receives a procedure for finding exception diacritics (indices), but does not know which particular diacritics will be used, or how many of them are necessary.

The second point is that hidden structure is sensitive to cues in the data (although these are cues in a different sense from Dresher and Kaye's 1990 model): when a data set can (or must) be represented with hidden structure, the absence of certain pieces of evidence in the data can lead to the grammar ignoring hidden structure. This situation arises in two chapters of this dissertation.

In chapter 2, the hidden structure learned consists of phonological features as well as of phonotactic constraints (see Hayes and Wilson 2008). It is demonstrated that, for phonotactic patterns that can be represented maximally concisely without the use of features, the grammars learned actually represent these patterns without the use of features (for instance, **[m]#* instead of **[labial,nasal]#*). Since features are the hidden structure learned by the model in chapter 2, this means that some constraints do not utilize hidden structure, even in instances where they could have.

In chapter 3, the learner is given access to a probabilistic version of Stratal OT (Bermúdez-Otero 1999), and the hidden structure learned consists of the intermediate

step(s) of a derivation. It is shown that the absence of evidence for the stratal affiliation of an opaque process (in this case, Canadian Raising; Joos 1942; evidence for stratal affiliation can come as evidence for non-interactive application of the opaque process, or as evidence that the opaque process is word-bounded, so that it cannot apply at the last derivational step) strongly decreases the chance that the process will be represented at the intermediate step, which is a word-bounded phonological stage. In the cases where the process is not represented at the intermediate step, learners choose to leave the intermediate step moot with respect to raising, which introduces unwanted free variation, as can be seen in section 3.3.3.

Before giving a more detailed overview of the projects pursued in this dissertation, I will first present the problem of hidden structure learning in more detail. Hidden structure can be defined as any structure that must be inferred by the learner, and cannot be directly observed in the learning data. In Optimality Theory (OT; Prince and Smolensky 1993), hidden structure is present whenever the learner does not know all the violation marks in a tableau based on the learning data provided, as will be demonstrated in (1) and (2) below. I will briefly exemplify the difficulty associated with learning hidden structure based on the two best-studied examples of hidden structure: metrical structure and underlying representations.

In the realm of metrical phonology (see, for instance, Liberman and Prince 1977, Dresher and Kaye 1990, and Hayes 1995), every metrical structure is only compatible with one unique stress pattern, but a single stress pattern might correspond to several metrical structures. For instance, Polish (Comrie 1967) has penultimate stress. In a three-syllable word like [tɛ.ˈlɛ.fɔn] ‘telephone’, this is compatible with a trochaic parse,

tɛ('lɛ.fɔ̃n), or an iambic parse, (tɛ.'lɛ)fɔ̃n.¹ This means that the violations of constraints such as All-Ft-Right (one violation for every syllable that intervenes between the right edge of the word and the right edge of some foot) and Trochee (one violation for every iambic foot) cannot be determined, as they differ between the two parses (see Jarosz 2013a for more discussion of this example):

(1) Violation marks for Polish stress

a. Learning data (no feet)

/tɛlɛfɔ̃n/	Trochee	Iamb	All-Ft-Left	All-Ft-Right
'tɛ.lɛ.fɔ̃n	?	?	?	?
tɛ.'lɛ.fɔ̃n	?	?	?	?
tɛ.lɛ.'fɔ̃n	?	?	?	?

b. With trochaic feet

/tɛlɛfɔ̃n/	Trochee	Iamb	All-Ft-Left	All-Ft-Right
('tɛ.lɛ).fɔ̃n		*		*
tɛ('lɛ.fɔ̃n)		*	*	
tɛ.lɛ('fɔ̃n)			**	

c. With iambic feet

/tɛlɛfɔ̃n/	Trochee	Iamb	All-Ft-Left	All-Ft-Right
('tɛ)lɛ.fɔ̃n				**
(tɛ.'lɛ)fɔ̃n	*			*
tɛ(lɛ.'fɔ̃n)	*		*	

A learner acquiring an analysis of the Polish stress system should be able to infer a trochaic system by comparing various lexical items; for instance, a two-syllable word like [lɛ.kaʃ] ‘physician’ would strongly bias the system towards a trochaic analysis. See Tesar (1998), Tesar and Smolensky (2000), Jarosz (2013a), and Boersma and Pater (2016) for some of the previous work on this problem in OT and related approaches.

Underlying representations (URs) are another instance of structure that must be inferred by the learner, since one and the same surface representation (SR) may be

¹ Some work suggests that metrical feet are timing units (see, for instance, Beckman and

compatible with several URs. For instance, in Russian, which neutralizes pretonic /o/ and /a/ to [ʌ] (depending on context; Halle 1959, Ward 1975, Timberlake 2004), the words [glʌ.'v-a] ‘chapter’ and [grʌ.'z-a] ‘thunderstorm’ have the same first vowel whenever their stem vowel is unstressed. However, the nominative plural of these words attracts stress to the stem (Stankiewicz 1993, Melvold 1989): [ˈgla.v-i] ‘chapters’ and [ˈgro.z-i] ‘thunderstorms’, respectively, which reveals a vowel contrast between the two stems. This contrast can only be accounted for if ‘chapter’ is given the UR /glav-/ and ‘thunderstorm’ is given the UR /groz-/, but this cannot be seen in the singular forms of these words. (2) below illustrates the uncertainty of Faithfulness violations [glʌ.'v-a]: if there is no UR, no Faithfulness violations can be assessed, whereas various possible URs lead to different violations of Faithfulness constraints. The Markedness constraint against [o] and [a] in pretonic position (represented as *_{O,a} / _'σ here) has the same violations regardless of the UR chosen, because Markedness constraints do not refer to URs.

(2) Violation marks

a. Learning data (no URs)

/???	* _{O,a} / _'σ	Ident(tense)	Ident(round)
[glʌ.'va]		?	?
[glo.'va]	*	?	?
[gla.'va]	*	?	?

b. UR with /o/

/glov-á/	* _{O,a} / _'σ	Ident(tense)	Ident(round)
[glʌ.'va]		*	*
[glo.'va]	*		
[gla.'va]	*		*

c. UR with /a/

/glav-á/	* _{O,a} / _'σ	Ident(tense)	Ident(round)
[glʌ.'va]		*	
[glo.'va]	*		*
[gla.'va]	*		

d. UR with /Λ/

/glΛv-á/	*o,a / <u> </u> 'σ	Ident(tense)	Ident(round)
[glΛ.'va]			
[glo.'va]	*	*	*
[gla.'va]	*	*	

Work that has addressed the issue of learning URs in OT and related approaches includes Johnson (1984, 1992), Tesar et al. (2003), Prince and Tesar (2004), Hayes (2004), Jarosz (2006a, b, submitted), Tesar (2014) and work cited there, Apoussidou (2007), and Pater et al. (2012).

In this dissertation, I will explore three types of structure that the learning literature usually does not treat as hidden structure: the phonological feature description of segments, intermediate derivational stages in serial grammars, and the marking of certain words in the lexicon as exceptions. These types of structures are essential to the learning of a phonological system, and as I will explain, they should all be seen as instances of hidden structure.

Features are essential for accounting for segmental phonological phenomena (Chomsky and Halle 1968). However, there are cases that show that assignment of phonological features to segments (for instance, whether [v] in a given language has the feature [sonorant]) is not always phonetically transparent, but, instead, language-specific (see, for instance, Anderson 1981, Mielke 2004, Hall and Žygis 2010; some examples are given in section 2.1.3.1). This means that the acoustic data a learner receives cannot be sufficient for an analysis in terms of features, even though features are necessary for phonological analysis. Thus, features should be treated as an instance of hidden structure – and, as Mielke (2004) suggests, they might be found by grouping together segments that participate in the same phonological pattern. While there have been studies in which

features are induced from acoustics (Lin 2005, Lin and Mielke 2008), learners that explicitly infer features from phonological context are at best understudied. This latter type of learner will be the focus of chapter 2 of this dissertation.

Intermediate derivational stages are a necessary component of any serial phonological grammar (Chomsky, Halle, and Lukoff 1956, Chomsky and Halle 1968, Bermúdez-Otero 1999, McCarthy 2008). However, the derivational precursors of a surface representation cannot be unambiguously derived from the shape of that surface representation. This is especially true in opaque interactions (Kiparsky 1973), where certain lexicon-wide generalizations made by the grammar are contradicted by the surface representation, but are true of an intermediate representation (see also McCarthy 1999). Thus, intermediate representations are unobserved, and since they play an important role in phonological grammars, they are an instance of hidden structure. Previous work that has explored the issue includes Jarosz (2015), and Staubs and Pater (2016). However, this work does not address the claims with respect to the relative learnability of various opaque interactions in Stratal OT (Bermúdez-Otero 1999, Kiparsky 2000) made by Kiparsky (2000). The learning of opaque interactions in a stochastic version of Stratal OT, with a special focus on the predictions made by Kiparsky (2000), will be the focus of chapter 3.

Finally, the marking of certain lexical items or morphemes as exceptions to lexicon-wide generalizations made by the grammar is necessary to capture any data set that has word-specific or morpheme-specific exceptions with respect to certain processes (see, for instance, Anttila 2002, Coetzee and Pater 2011). In order for a grammar to be consistent, it needs to know which forms should be treated as exceptions to the logic

encoded in the grammar, but exceptionality is not explicitly marked in the data as an infant encounters it, so that it would not be fair to mark it in the training data for a machine learning. Thus, the induction of exception marking, which will be the focus of chapter 4, is a form of hidden structure.

Acknowledging the fact that features, derivational stages, and exceptionality marking are types of hidden structure makes the learning problem much more difficult, because the learning space is much larger: the learner's hypothesis space contains grammars that give segments every possible feature assignment, grammars that give surface representations every possible derivational history, and grammars that give lexical items every possible set of exception markings for every process in the language's phonology.

The rest of this dissertation consists of three chapters that demonstrate that this additional but necessary difficulty can be handled by extending currently available learning techniques. Chapter 2 is a revised version of "A radically emergentist approach to phonological features: implications for grammars" (previously published in *Nordlyd* 41(1):21-58), and is an exploration of learning feature labels for phonological segments. This chapter demonstrates that, given a learning procedure that prefers generality in grammars, and has no phonological features given *a priori* but needs to induce these features, grammars that use both feature labels and segment labels (see, for instance, Gallagher 2013) are preferred to grammars that only use feature labels to describe segmental content (see, for instance, Chomsky and Halle 1968).

Chapter 3 is a revised version of "Learning opacity in Stratal Maximal Entropy Grammar", co-authored with Joe Pater and currently under review for journal

publication, and demonstrates the learnability of opaque interactions in a stochastic version of Stratal OT (Bermúdez-Otero 1999, Kiparsky 2000) implemented in Maximum Entropy Grammar (Goldwater and Johnson 2003). Our case studies also suggest that opaque interactions are learned less easily than their transparent counterparts, but evidence for the stratal affiliation of the opaque process can aid the learnability of opaque interactions (Kiparsky 2000). My contribution to this paper consists of co-designing, running, and interpreting the simulations, as well as writing sections 3.2-3.5.

Finally, chapter 4 proposes an algorithm for inducing and ranking lexically indexed constraints (Kraska-Szlenk 1995, Pater 2000, 2010), and applies it to the problem of learning Dutch main stress assignment (Kager 1989, Nouveau 1994, van Oostendorp 1997). Dutch main stress assignment is governed by a Quantity-Sensitive (QS) rule (Nouveau 1994, Ernestus and Neijt 2008, Domahs et al. 2014), but also exhibits widespread exceptionality. I show that the proposed learner is both able to find an appropriate regular grammar, and account for the exceptions to this regular grammar.

The projects described in these three chapters all attempt to open a window into a new type of hidden structure. However, much work remains to be done in the future. For all three projects, work is needed that would further connect the results of the model to novel data. In chapter 2, there is a need to give the feature classes learned by the model a phonetic definition, so that they can be generalized to novel segments; this will allow for the results of the model to be tested in an behavioral setting (see section 2.5). In chapter 3, it is not clear what happens when the learner fails to produce the appropriate analysis for a data set with an opaque interaction. The learner by itself predicts free variation between different outcomes, but it is not clear if this would lead to morphologization of

the process (see, for instance, Bybee 2001) if a morphological module had been present in the model; this is an important topic for future work. Finally, the model in chapter 4 learns the divide between exceptions and rule-obeying forms in Dutch, so that an analysis without a default grammar, along the lines of that provided by van der Hulst (1984) and Nouveau (1994), comes out. However, on nonce word tests, speakers show some influence of exceptional forms on how they stress nonce words (see section 4.2.1.3). This means that a bridging model should be built that might generalize the exception indices found by the learner to novel items, based on various phonological (and perhaps other) similarity with existing words in each exception class – which is similar to the extension necessary for the model developed in chapter 2.

Apart from these necessary extensions, all three models should be tested on other case studies to ensure the generalizability of the conclusions presented. For the model in chapter 4, it would be particularly interesting to test the model on a case that includes both within-word variation and exceptionality. For the other two models, more elaborate case studies should be used in the future to demonstrate their ability to learn features and opaque processes, respectively.

It would also be interesting to see the behavior of a model that combines and integrates the hidden structure learning mechanisms considered here. In particular, integrating the learning of segments into the feature learning model in chapter 2 would be interesting, but it would also be very worthwhile to consider a model that combines learning opacity (as in chapter 3) and learning exceptions (as in chapter 4).

Finally, since phonological analysis does depend on a large number of variables that are not designed to be directly observable in the data that a language-learning infant

encounters, more types of hidden structure learning should be addressed. For instance, the induction of prosodic units such as feet and morae (i.e., the grammar starts out without these prosodic units, and induces them as they become necessary) would be a very interesting topic for future work. Through this work on hidden structure, we can not only find out more about the learnability of existing phonological frameworks, but we can also discover what kind of phonological abstractions are supported by data from a particular language. This bottom-up view of phonological structure can provide interesting views on the debate (cf. Blevins 2004, 2006 vs. Kiparsky 2006, De Lacy and Kingston 2013) on what we must assume is innate in phonological representation and grammar, and what can be assumed to be learned from the data.

CHAPTER 2

THE EMERGENCE OF REDUNDANCY BETWEEN FEATURES AND SEGMENTS

This chapter is a revised version of “A radically emergentist approach to phonological features: implications for grammars”, published in *Nordlyd* 41(1), 21-58.

2.1. Introduction

In this chapter, I will introduce a novel way of looking at models of emergent phonological structure (in particular, models of emergent phonological features). I will argue that such models can make predictions not only about typology (Blevins 2004, Mielke 2004), but also about the shape of grammars of individual languages.

I will propose a radically emergentist (Blevins 2004, Wedel 2003, 2011, Mielke 2004) model of learning phonological features: the question posed is what the function of phonological features will be in the grammar when the learner starts with no features, and only has a drive towards generalization and a way of creating features. Thus, features emerge from the satisfaction of the learner’s desiderata, but are not a required element. This makes it possible to test where phonological features are necessary and where they are not.

The learner proposed here induces features for phonological segments based on their grammatical patterning (cf. Mielke 2004). Inducing feature labels based on this kind of information is necessary given the available evidence for language-specific phonological classes (see section 2.1.3.1 below). This method is similar to other hidden structure inference methods (e.g., those for metrical feet and underlying representations, see chapter 1).

One simplifying assumption that will be made, however, is that the acoustic and articulatory aspects of these segments will be left out of consideration, despite the evidence that phonological features do have phonetic content (see, for instance, Saffran and Thiessen 2003, Cristià and Seidl 2008, Cristià et al. 2013, Kuo 2009). This simplifying assumption is only made because there are no pre-existing phonological feature learners that learn from phonological patterning, and building in acoustic and articulatory dimensions into the model would have made it more difficult to implement. In this manner, the model proposed here can be seen as a pilot that is in need of expansion with a phonetic module.

Through computational simulations, it will be shown that this emergentist model predicts grammars that diverge from what is standardly assumed. In the resulting grammars, phonotactic constraints do not always refer to segmental information through phonological features, but only do so when this helps the grammar achieve a shorter and more general description of the grammar (cf. the literature on Minimal Description Length, see, for instance, Rasin and Katzir 2016 and work cited there). If the grammar indeed may refer to the same sound either on the level of segments or on the level of features, this has interesting implications for the plausibility and learnability of multi-level phonological representation, along the lines of those proposed by Goldrick (2001), Bye (2006), Boersma (2007), and van Oostendorp (2008), among others.

However, before addressing these issues, I will first briefly review emergentist models of linguistic structure in section 2.1.1, after which section 2.1.2 will review how features increase generality and therefore an (indirect) drive towards generalization motivates the presence of features in the current proposal, section 2.1.3 will show which

tools the learner is given to induce features, and section 2.1.4 will describe the grammar learning model used.

Section 2.1.5, then, will address how this learner is tested on a simplified test case from English phonotactics and give a preview of the results and their implications. Finally, section 2.1.6 will give an overview of the rest of the chapter.

2.1.1 Emergentist models of linguistic structure

Since the 2000s, the idea that various kinds of phonological structure may be “emergent” (see, for instance, Blevins 2004, Wedel 2003, 2011, Mielke 2004) has gained traction in the literature. These ideas depart from the canonical hypothesis that most or all elements of phonological structure are innate and universal (see, for instance, Chomsky and Halle 1968 for universal phonological features). When some aspect of phonological structure is emergent, I take this to mean that it is not necessary to stipulate the particulars of that aspect (such as individual phonological features) in Universal Grammar – which is similar to the approach taken by Dresher (2013, 2014).

The debate on whether certain aspects of phonology are emergent or innate (see, for instance, Blevins 2006 and Kiparsky 2006 for a debate on the emergence of final devoicing and the rarity of final voicing) has mostly centered on typological facts. In the realm of phonological features – which will be the focus of this chapter – Mielke (2004) motivates his Emergent Feature Theory by appealing to the typology of phonological patterns: no universal feature theory has a good account of the range of phonologically active classes in his cross-linguistic database (Mielke 2007). Instead, he proposes an Emergent Feature Theory: phonological patterns start out in the phonetic realm, and are gradually extended to phonetically similar segments by analogy. When these patterns are

interpreted phonologically, all segments participating in the pattern that resulted from this phonetic analogy is assigned a phonological feature label by virtue of their participation in the pattern alone. Thus, the kind of phonological feature that emerges in this model is abstract and based on phonological distribution, while phonetic content comes from analogy in the phonetic history of phonological patterns.

This model has the potential to capture the tendency of languages (and speakers) to have processes apply to phonetically coherent groups of segments (because of phonetic analogy), while allowing for generalizations along phonetic dimensions that are not captured by any single feature theory. At the same time, this model allows for processes that apply to phonetically disjoint groups of segments, such as the ones cited by Anderson (1981) and Mielke (2004) (and briefly exemplified in section 2.1.3.1), since features are induced based on distributional information alone.

This general scenario is the inspiration for the model that will be proposed here. However, instead of viewing the induction of features as simply a requirement for accepting a pattern into the phonological grammar (because of the assumption that phonological grammars must be formulated in terms of features, shared by both Chomsky and Halle 1968 and Mielke 2004), I propose here that the motivation for learning features is the implicit desire of the language-learning infant to induce grammars that have maximally general constraints for every phonological pattern. I will demonstrate that this leads to an interesting outcome: features are consistently desired for multi-segment patterns only.

2.1.2 Features as a way to increase generality

2.1.2.1 The necessity and use of features in the grammar

As pointed out by Halle (1978), the formulation of a grammatical statement (rule or constraint) in terms of phonological features means that this statement will be applicable to all segments with that feature description, including novel segments. Halle gives the example of the name Bach [bax], which, despite ending in a non-English consonant [x], still triggers devoicing of the suffix [-z]: /bax+z/ → [baxs], *[baxz] “Bach’s/Bachs”. This provides evidence for the idea that the rule of devoicing [-z] refer to the feature [(±)voice], and that [x] is classified with respect to that feature. This methodology is called “Bach-testing” (Halle 1978 credits the idea to Lise Menn). Other tests (such as wug-testing; Berko 1958) have also shown that language users generalize rules of mental grammar to new tokens.

Based on this evidence, I follow Chomsky and Halle’s (1968) and Albright and Hayes’ (2002, 2003) ideas about generalization as a driving force in the acquisition of grammar. In fact, I will assume that finding the most general formulation of a phonological pattern is the main driving force behind learning phonological grammars.

If a grammar does not have access to classificatory phonological features, it will miss many generalizations – every pattern which applies to more than one segment will have to be triggered by a series of constraints. For instance, a process of final devoicing (as, for instance, in Dutch – see Booij 1995) will have to be triggered by constraints that refer to each of the individual segments undergoing final devoicing: *b#, *d#, *v#,

Since features help the learner reach generality in grammar (as was explained above), it can be said that the presence of features is motivated by the drive towards

generality rather than an *a priori* requirement. In other words, the learner is set up such that it could reach full accuracy on the data without phonological features, but it is the pressure to state phonological patterns as broadly as possible that makes the learner induce these features, as will be explained in sections 2.1.2.2 and 2.1.2.3 below.

2.1.2.2 Motivating the presence of features through generality

The learner introduced in this chapter motivates the presence of features in the grammar by the fact that they make the grammar able to express the same pattern with fewer constraints, ultimately creating a more general grammar (cf. section 2.1.2.3). This is in line with the motivation for the presence of features cited in Chomsky and Halle (1968) and Kenstowicz and Kisseberth (1979): they make it possible to formulate rules in a more general and concise way. McCarthy (1981b) confirms the adequacy of such an approach. However, Chomsky and Halle and subsequent work assume that features are present in any rule *a priori*. Much of the existing work that induces features (see, for instance, Lin 2005, Lin and Mielke 2008) also assumes that phonological categories such as features are learned because there is an *a priori* (innate or otherwise) requirement to have such categories (the grammar requires the presence of phonological features, for instance).

The model proposed here, however, maintains that the presence of features is motivated by an external factor – namely, the goal of having maximally general constraints in the grammar. Since I assume that the linguistic representation of segmental content (which we may reasonably assume must take the shape of segment and feature labels) is not innately specified and thus subject to between-language variation, the constraints that are part of grammars cannot be universal (as is standardly assumed in

Optimality Theory (OT) – see, for instance, Prince and Smolensky 1993). If phonological constraints are based on phonological representations, and if the building blocks of phonological representations (segments/features) are not fixed before the reception of language input, then neither can phonological constraints be fixed before language input is received. For work discussing and implementing the induction of phonological constraints, see, for instance, Hayes (1999), Hayes and Wilson (2008) and Wilson (2010).

In the model proposed in this chapter, induction of phonological constraints and induction of phonological features interact in a cyclic way: the induction of phonological constraints prompts the induction of features, and the induction of features prompts the induction of new constraints. I will explain the details of this interaction in section 2.2.2. It is this cyclic interaction that leads to a gradual increase in generality by gradually recruiting phonological features into the grammars. This interaction also leads to the general effect that phonological features are only recruited when necessary – which is precisely the property highlighted by the case study taken up in this chapter, which I will now describe.

2.1.2.3 Implementing generality

The way in which the drive towards generalization is implemented in this model is different from other approaches. Albright and Hayes (2003), and Calamaro and Jarosz (2015) implement this drive by selecting maximally functional rules into the grammar (with “functional” being defined by specific metrics that have to do with the scope and reliability of the rules). On the other hand, Rasin and Katzir (2016) have generalization follow from their goal of Minimal Description Length.

In the current model, building on the proposal in Wilson (2010), generalization is

obtained through constraint selection by information gain (see section 2.3.2.1.2), and through regularization in the optimization procedure (see section 2.3.2.5). These are both general purpose mechanisms that are useful for independent reasons. The fact that they can yield a drive towards generalization as a byproduct means that no dedicated mechanism for generalization is necessary. In particular, the goal of Minimal Description Length (see Rasin and Katzir 2016 and work cited there) appears to converge with the effects of information gain and regularization, at least for the case study in this chapter.

Information gain estimates how much a constraint will help the grammar attain a closer match to the training data – see (22) in section 2.3.2.1.2 for a formal definition. This measure is proposed by Della Pietra, Della Pietra, and Lafferty (1997) and Wilson (2010) as a constraint selection mechanism, since constraints that are relevant to the data set will be able to bring the grammar closer to matching the data set than constraints that are irrelevant to the data set.

However, it is also the case that, among the constraints that are relevant to the data set, more general constraints will be able to help the current grammar match the training data more closely simply because they encompass more forms. Banning a large number of ungrammatical forms in one constraint means that that constraint will be able to bring the grammar closer to its goal of allowing all and only the training data than a constraint that bans a smaller number of ungrammatical forms. In this way, information gain also helps the model to select general over specific constraints, pushing the grammar towards generalization.

Regularization is an extra term in the objective function that biases the model to minimize constraint weights (either their sum, as in an L1 prior, or the sum of squares, as

in an L2 prior). Terms of this type are often used to prevent batch optimization procedures from letting weights become so high that they approach infinity. However, regularization also makes sure that an optimally sparse solution to a problem is found. One of its possible effects is that as few constraints as possible are employed in the analysis, each with the smallest degree of importance (weight) possible (see section 2.3.2.5 for more discussion). Along with information gain as a constraint selection criterion, regularization is a non-specific and indirect way of pressuring the grammar towards generality. Regularization is often needed for optimizing constraint weights as a technical aid (to avoid the weights' approximating infinity), so that it is not present specifically to enhance the grammar's generality. Regularization also does not appeal to any specific parts of constraints – it only tries to lower the weight of constraints whose function is already covered by another, more general constraint (for instance, if there is a constraint *#[m], and a more general constraint *#[labial], then the latter constraint covers the function of the former constraint, and the weight of *#[m] will be pushed down by regularization).

Thus, generality is defined in terms of function, not form (cf. Albright and Hayes 2003, Calamaro and Jarosz 2015, and Rasin and Katzir to 2016): a grammar becomes more general as a by-product of regularization, which is needed for independent reasons, rather than being stipulated by a dedicated principle. This is in line with the emergentist theme in this chapter: I am interested in what will happen when ingredients that are motivated by independent factors (regularization is motivated by the need to keep constraint weights finite) are used to provide a pressure to induce phonological features.

A separate set of test data to measure the generality in the grammars found by the

learner was not included. This is because I am interested in whether the grammar will represent certain generalizations with segments or features, and this information can be directly seen in the representations of the grammars. However, future versions of this learner should also explicitly test the grammar on novel segments, along the lines of what is suggested in sections 2.2.4 and 2.5.

2.1.3 Why features are hidden structure and how to learn them

2.1.3.1 Classificatory phonological features: not always phonetically transparent

The learning setup chosen here crucially depends on the distinction between classificatory and phonetic features (Chomsky & Halle 1968). Classificatory features are entities that classify the segment (allophone or phoneme) categories of a language. In other words, (classificatory) features are maps from groups of segment types to the presence of a category label on those segments, e.g., /p, b, m/ → [(+)labial].

On the other hand, Chomsky and Halle (1968) define phonetic features as real-valued articulatory or acoustic dimensions. Phonetic features can be seen as maps from segment tokens to values on a phonetic dimension, e.g., [m] → [0.9 labial closure], [t] → [0.5 labial closure].

In this chapter, I will focus on the formation of classificatory features, and, in fact, I will use the term “feature” when used without further qualification as a shorthand for “classificatory feature” in the remainder of the chapter. The content of classificatory features may be identical to some phonetic dimension, as is essentially assumed by Chomsky & Halle (1968), but since phonetic and classificatory features are distinct concepts, this is not necessary. In fact, there is evidence that, at least in some cases, classificatory features can be phonetically arbitrary, and need to be induced from some

non-phonetic factors.

Anderson (1981) and Mielke (2004) show that there are classes of segments referred to by phonological grammars that do not correspond to a single phonetic dimension, or to an intersection of phonetic dimensions. For instance, Mielke (2004) reports that in Evenki (Nedjalkov 1997), the segments /v, s, g/ undergo nasalization at the beginning of a suffix and after a nasal. Similarly, in Kolami (Emeneau 1961), the choice between [-l] and [-ul], allomorphs of the same plural marker, is made based on whether the preceding segment is a member of the set /t, d, ɳ, r, l, i, e, a/, to the exclusion of /p, t̪, k, ɖ, g, s, v, z, m, ŋ, j/. Under the standard logic that segments that participate in the same phonological pattern should share a feature label, this means that at least some classificatory features must exist that do not fully correspond to any particular phonetic dimension.

There are also cases in which phonetically very similar segment types are classified differently in different languages. For instance, Hall and Žygis (2010) show that [v] is classified as a sonorant in some languages, while it does not fall in the class of sonorants in other languages.

Similarly, implosive stops like [ɓ, ɗ] appear to differ cross-linguistically as to whether they are interpreted as glottalized (presence of [constricted glottis]), and/or whether they are obstruents or sonorants (absence or presence of [sonorant]). In Maidu (Shipley 1964), implosives pattern as non-glottalized obstruents. There are plain stops, ejectives, and implosives, but ejectives and the glottal stop are disallowed at the end of a morpheme, while implosives are allowed there; in addition, ejective [p', t'] have a highly restricted distribution in two-consonant clusters, while implosive [ɓ, ɗ] do

not. This suggests that implosives do not have the feature [constricted glottis], but the contrast between [p, t] and [ɓ, ɗ] is made by a feature like [voice]. [ɓ, ɗ] also participate in a devoicing process that turns them into plain stops [p, t], while voiced nasals turn into (partially) devoiced nasals. If this process simply takes away the feature [voice], this means that [ɓ, ɗ] must be obstruents.

In Lua (Boyeldieu 1985), on the other hand, implosives pattern as glottalized sounds. First, implosive stops [ɓ, ɗ] contrast with plain, voiced, and prenasalized stops [p, t, b, d, ^mb, ⁿd], and [ɓ, ɗ] turn into preglottalized nasals [ʔm, ʔn] rather than plain nasals [m, n] before a nasalized vowel (Boyeldieu 1985:154). Furthermore, [ɓ, ɗ] also contrast with both oral and nasal sonorants [w, l, j]), which turn into [m, n, ɲ] before a nasalized vowel. This shows that the most logical way of making the contrast between [b, d] and [ɓ, ɗ] is through [constricted glottis]. Furthermore, there are some hints that [ɓ, ɗ] might be obstruents: all sonorants occur word-finally, and most sonorants occur word-medially, whereas [ɓ, ɗ] only occur word-initially (unless it is at the beginning of the second member of a compound), and most obstruents also occur only word-initially. However, this might also be accounted for by the glottalized nature of [ɓ, ɗ]: the glottal stop, [ʔ], is also restricted to word-initial position.

Finally, in Ebrié (Ahoua 2009, Clements 2000), the implosive [ɓ] behaves like a sonorant (and there is no evidence for glottalization). There is a process by which [ɓ, l, j, w] turn into their nasal counterparts [m, n, ɲ, ɳ] after a nasalized vowel. This process does not apply to any obstruents, including plain, aspirated, and voiced stops, and voiceless and voiced fricatives. Since there are no other glottalic consonants in the language, it is impossible to evaluate whether [ɓ] is assigned the feature [constricted

glottis], but the specification [labial, sonorant] (without [nasal]) is sufficient to contrast [6] with every other consonant in this language.

Thus, there is evidence that at least some segment classes in languages are defined by features that do not correspond to a phonetic dimension, and there are segments that seem to have different featural interpretations per language. This means that at least some phonological feature specifications in languages cannot be determined before the data has been analyzed. In other words, in terms of learning, features have to be treated as hidden structure (see also chapter 1).

2.1.3.2 Learning classificatory features from phonological patterning

In learning the hidden structure problem thus posed by classificatory features, I will follow the logic that a segment that participates in phonological pattern P shares a label with the other segments that participate in that same phonological pattern P (cf. Mielke 2004). By this logic, the analysis required for finding the correct feature specification for a segment involves finding phonological patterns, and finding which segments participate in each pattern.

In the learning model proposed here, phonological patterns are found by selecting phonotactic constraints with information gain (see section 2.3.2.1.2), while segments that participate in each pattern are found by clustering over segments' information gain values when inserted in specific constraint contexts (see sections 2.3.2.2-2.3.2.3). These clusters of segments are then labeled with arbitrary feature labels. The subsequent use of the feature labels in the grammar is encouraged by regularization (see section 2.3.2.5), which provides an indirect pressure towards generalization.

Together, these elements in the learner lead to a phonotactic Maximum Entropy

grammar (Hayes & Wilson 2008) in which the constraints themselves, as well as the featural representations necessary for formulating these constraints, have been induced by the phonological learner. The phonetic aspect of these features will be left outside of consideration simply to concentrate on building a learner that is able to extract phonological features from phonological patterning; however, it is essential to have a phonetic component to features, and the most important future goal for this project is to add such a phonetic component to the model.

Since classificatory features are maps from groups of segments to labels, segments have to be in place before induction of classificatory features can occur. I will assume that, before induction of classificatory features, segments (segment types) have been learned from acoustics, along the lines of Elman and Zipser (1988), Lin (2005), McMurray, Aslin, and Toscano (2009), and Goldsmith and Xanthos (2009). See also McQueen, Cutler, and Norris (2006) and Nielsen (2011), among others, for evidence that individual segments are psychologically real. To concentrate on learning features from phonological patterning, the segment learning step will be omitted, but segment and feature learning should be integrated in future work.

The division of learning phonological representations into two phases (segments first, features after) is in line with work such as Pierrehumbert 2003a, Peperkamp et al. (2006), and Calamaro and Jarosz (2015), which exemplifies a two-stage approach to learning. In the case of Pierrehumbert (2003a) and Peperkamp et al. (2006), low-level abstract representations (e.g., phones) are learned in a bottom-up fashion, and higher-level abstract representations (phonemes) are learned based on these low-level abstract units. In the case of Calamaro and Jarosz (2015), rules are constructed over segment

pairs, while phonological features are used to generalize rules that span several segment pairs.

Of course, it is not obvious that the learning of phonological, classificatory features takes place after segment categories have been acquired. It may be the case that classificatory features are learned simultaneously with segment categories – see, for instance, Dresher (2014) for discussion of this scenario. However, the scenario explored here, even if not obviously correct, at least has given fruitful results in previous work.

There has been a good amount of work on learning segments from acoustics (see, for instance, Elman and Zipser 1988, Niyogi 2004, Lin 2005, Vallabha et al. 2007, McMurray, Aslin, and Toscano 2009, Goldsmith and Xanthos 2009, Elsner et al. 2013, Elsner, Antetomaso, and Feldman 2016, Dillon, Dunbar, and Idsardi 2013, and Boersma and Chládková 2013). However, the learning of classificatory features from the phonological behavior of segments, as outlined in Mielke's (2004) Emergent Feature Theory, has not been explored as much (except for Archangeli, Mielke, and Pulleyblank 2012). For this reason, I chose to focus on this aspect of learning.

In future work, learning the phonetic aspect of features should be integrated into the model (see Lin 2005, and Lin and Mielke 2008 for work along these lines). There are at least two ways in which phonetic factors could be incorporated into the feature learning module. First, there could be a bias towards creating acoustically and/or articulatorily consistent classes of segments (although, as pointed out in section 2.1.3.1, there can be classes of segments that are not consistent acoustically). Second, once features are induced, then, for every feature [F], the acoustic (and/or articulatory) traits that unite all segments that carry [F] can be found, and attached as the phonetic correlates

of [F].

When phonological features are combined with phonetic correlates, the classes defined by phonological features become open-ended, so that Bach-testing (as in section 2.1.2.1) becomes available. This, in turn, allows for the main findings of this study (see sections 2.1.5 and 2.4) to be tested experimentally (see section 2.5).

Induction of segments (Elman and Zipser 1988, Lin 2005, Niyogi 2004, Vallabha et al. 2007, McMurray, Aslin, and Toscano 2009, Goldsmith and Xanthos 2009, Elsner et al. 2013, Elsner, Antetomaso, and Feldman 2016, Dillon, Dunbar, and Idsardi 2013, and Boersma and Chládková 2013) should also be integrated with the current model. For such integration to happen, the learner should start with raw acoustics, then learn segment categories from these, and then use both acoustics and the patterning of segment categories to find features.

However, in the current model, I focus only on the induction of classificatory feature labels from the phonological patterning of segment labels, as a first step towards a fuller model along the lines described in the paragraphs above.

2.1.4 Learning grammars: phonotactic constraint induction and weighting

Since the current model is a model of inducing and recruiting feature labels in the process of creating a maximally general grammar, there must be a component that learns grammars. Because features are not given at the outset, while it is expected that some constraints in the grammar will refer to features, the constraints in the grammars cannot be universal and must be induced as well. I will follow Hayes and Wilson's (2008) and Wilson's (2010) Maximum Entropy approach that induces phonotactic constraints in addition to finding optimal weights for these constraints. Thus all information used by the

feature learner (briefly described in section 2.1.3.2 above) will be phonotactic.

Joint induction of grammar and features has been explored at least in one instance: Archangeli, Mielke, and Pulleyblank (2012) explore a model that induces grammatical constraints that are allowed to refer to sets of segments. An example of this is the (positively formulated) vowel harmony constraint $\{i, u\} \rightarrow \neg\{e\}$: if the first vowel is one of $\{i, u\}$, then the second vowel is not a member of the set $\{e\}$. A somewhat similar model is given by Goldsmith and Xanthos (2009): segments are induced from acoustics, and then the model estimates whether the language has vowel harmony based on these segments.

However, the difference between Archangeli, Mielke, and Pulleyblank's (2012) model and the model presented in this chapter is that the former model does not seek to explicitly label the sets of segments that occur in a generalization, whereas the current approach does induce labels for sets of segments. Because the current model finds labels for sets of segments that occur in generalizations, my approach can make the difference between [labial, nasal] and [m] (if [labial] and [nasal] are labels for sets of segments that group together based on phonological behavior), even when the representations [labial, nasal] and [m] stand for the same sound ([m]). It is this property that allows the current model to gauge the usefulness of a segment representation ([m]) versus a feature representation ([labial, nasal]) of the same sound.

Another computational model that allows reference to segment identity as well featural generalization is presented by (Colavin, Levy, and Rose 2010). However, in this model, segmental identity is only available in OCP-like constraints (specifically, constraints that ban identical consonants in a certain domain), and features are not learned

by the model. By contrast, the current model introduces the possibility of referring to segment identity in any (phonotactic) constraint, and it also allows for learning the content of features (the necessity of which is motivated in section 2.1.3).

2.1.5 Testing the model and results prediction

The model of learning grammars and features just discussed, which I call radically emergentist, is implemented computationally, and applied to a simplified version of English phonotactics. The particular English patterns that are of interest are the following three:

(3) Three phonotactic patterns in English

a. Three-consonant clusters consist of [s] followed by a stop and a liquid

[stre] “stray”	*[ftre], *[ntre]
[splæʃ] “splash”	*[fplæʃ], *[mplæʃ]

b. No two stridents in a row

/bʊʃ/ + /z/ → bʊʃz	“bushes”	*[bʊʃs]
/bætʃ/ + /z/ → bætʃz	“batches”	*[bætʃs]
but: /mov/ + /z/ → movz, *moviz	“mauves”	

c. Three-consonant clusters must end in [t, θ, s] (Jensen 1993)

[nɛkst] “next”	*[nɛksp]
[sɪksθ] “sixth”	*[sɪksf]
[mʌmps] “mumps”	*[mʌmpf]

The crucial property of this data set is that the pattern in (3a) applies to the single segment [s], which is at the intersection of the class of sounds that participates in (3b) and the class of sounds that participates in (3c). This is interesting because this means that the learner has the opportunity to induce two classes of sounds ([s,z,ʃ,ʒ,tʃ,dʒ]=[F] for (3b), and [t,θ,s]=[G] for (3c)), and define the single segment [s] in pattern (3a) as the intersection of those two: [s] = [F,G]. This provides a test for whether the learner briefly introduced in sections 2.1.2-2.1.4 will represent pattern (3a) with a constraint *#^[s]CC (no three-consonant initial clusters that do not start with [s]), or with a constraint

*#[^][F,G]CC (no three-consonant initial clusters that do not start with a segment labeled [F,G] – which includes [s] and, potentially, other segments).

To extract the bare basics of this problem and provide a maximally simple and transparent test case for the learner, a toy language was constructed that has the three phonotactic restrictions in (4). While the learner was tested on this toy language to make the problem more concise, there are no inherent reasons why the learner could not be applied to the actual English case study.

(4) phonological patterns in the toy language

a. no [m] word-finally

baman, bamab; *bamam

b. no nasals word-initially

baman; *maman, *naman

c. no labials in between high vowels

baman, binin; *bimin, *bibin

Similarly to the English case, pattern (4a) can be expressed either in terms of the phone [m] alone – as the constraint *m# – or in terms of the intersection of the two classes used in (4b,c): *[labial, nasal]#. The standard, innate feature approach predicts that the constraint *[labial, nasal]# will be chosen to represent this pattern. However, I will show that my model predicts that the constraint *m# will be chosen.

This constraint, *m#, which appeals to linguistic sound without the mediation of classificatory features, co-exists with other constraints that do appeal to classificatory features in the grammars that are learned by the radically emergentist model. In other words, my model leads to grammars in which some constraints refer to sound through features, but other constraints refer to sound through a lower-level type of representation: segments (phones). This means that the grammars generated by the emergentist model

refer to multiple levels of phonological abstraction.

This latter finding is at odds with the usual assumption that all constraints in phonological grammars are encoded in terms of features. The idea that grammars only refer to features is a natural consequence of the assumption that phonological features are the innate units of classifying linguistic sound for the purposes of grammar (Chomsky and Halle 1968). Even models that do acknowledge that the processing of linguistic sound may take place at various levels of abstraction (acoustics, segments, features, ...) usually maintain that the phonological grammar itself refers to phonological features only (see Pierrehumbert 2003b).

The learner proposed here, however, generates grammars that have constraints that refer to different levels of abstraction for the same sound. In the toy language, for instance, a sound fragment which sounds like [m] may be referred to in one constraint by the feature combination [labial, nasal], and in another constraint by a featurally unanalyzed label “m”.

The emergent feature learning scenario proposed here makes predictions that are different from standard assumptions about grammar. Instead of only appealing to features, the grammars predicted by this model sometimes refer to atomic segment units, and sometimes to features. Some consequences of this will be seen in sections 2.4 and 2.5, but the crucial difference between this state of affairs and canonical assumption about grammars is that atomic segment units (e.g., [m]) are representational elements separate from features, and not simply shorthand for some feature bundle, whether this feature bundle be the same across languages, as in SPE (Chomsky and Halle 1968) for example, or potentially different between languages (Mielke 2004).

It is in this sense that the model of emergent features which will be presented here makes predictions about within-speaker grammatical structure. Some (non-prosodic) constraints in the grammar are predicted not to be encoded in terms of feature bundles, but in terms of atomic, featurally unanalyzed labels denoting segment categories. This has important conceptual consequences, but testable empirical predictions can also be derived from this: section 2.5 will sketch these predictions. However, the main result obtained in this chapter is that grammars appealing to a mix of features and atomic segments are a natural consequence of the model described here.

2.1.6 Chapter preview

The rest of this chapter is organized as follows. Section 2.2 will present my theory of phonological category emergence in more detail, and will also describe in more detail the toy language sketched above and the intuitions as to the consequences of the model as applied to the toy language. After this, section 2.3 will describe the computational implementation of the model applied to the toy language, and section 2.4 will show the results of this simulation. Finally, section 2.5 will offer discussion and concluding remarks.

2.2. A radically emergentist model of grammar and feature learning

2.2.1 Classificatory phonological features as byproducts of grammar learning

The model that will be proposed here is one that jointly induces phonological grammars and phonological features. This is inspired by inductive perspectives on the learning of both phonological features (Mielke 2004) and phonological grammars (Hayes 1999, Hayes and Wilson 2008, Wilson 2010). However, the synthesis made here is novel: I will assume a minimum of prior knowledge of representations – since the learner is not

given any phonological features in advance, and the way towards finding phonological features is dictated by the grammar.

As shown in work like Linzen and Gallagher (2014, under review), Calamaro and Jarosz (2015), Rasin and Katzir (2016), and Prickett (in progress), phonological features enable a more general statement of phonological patterns, and thus, feature-based constraints are generally preferred. A feature such as [voice], if assigned to voiced obstruents only, helps state the generalization that Dutch does not allow word-final voiced obstruents (Booij 1995) in one single constraint (for instance, *C[voice]#). If there is no feature [voice], then the same generalization should be expressed by banning each single voiced phone in Dutch separately (*b#, *d#, *v#, *z#, etc.), which lacks a general acknowledgment of the pattern of final devoicing (in terms of traditional phonological analysis, this would be a “missed generalization”).

As pointed out in section 2.1.3, the type of phonological feature appealed to here is the classificatory phonological feature (Chomsky and Halle 1968) – a unit that generalizes over individual segment (allophone or phoneme) categories. Chomsky and Halle (1968) do not posit an explicit level of segments. However, they do acknowledge the + and – in classificatory features (for instance, [+ voice] or [-voice]) must be mapped onto the continuous scale of phonetic features (for instance, [0.6 voice] or [0.4 voice]). These correspondences are expressed in contextual rules (e.g., [+ voice] → [0.9 voice] / __ [+ vocalic]).

In the approach taken here, where classificatory features are learned from the phonological behavior of segments, the function of mapping from continuous phonetic features to categorical classificatory features is performed by segments. Because of

classificatory features can be phonetically non-transparent (see section 2.1.3.1), features are based on and induced from groupings of segments that are active in a certain context, as shown in **Figure 1**. Phonetic factors should ideally play a role in the definition of classificatory features, but, as was already expressed in section 2.1.3, in this model, phonetic factors are simplified away in favor of the other aspect of classificatory features, namely, acknowledging groups of segments that undergo the same process.

Figure 1. The relation between phonetic and classificatory features

Classificatory features (categorical)



Segments



Phonetic features (gradient)

One potential objection to this setup comes from Chládková (2014), who argues that, in the perception of vowels, F1 is mapped directly onto a categorical height feature, instead of onto phoneme categories that are then classified in terms of features. A vowel discrimination task with Czech speakers was used to show that vowel height is discriminated equally categorical both in the regions of the vowel space where existing phoneme categories are canonically realized, and in the central region of the vowel space, where Czech does not have canonical realizations of phonemes. Since speakers are shown by Chládková (2014) to have difficulty identifying the phonemic identity of vowels in the central region vowel space, discrimination should not be categorical in this region of the vowel space if perception of F1 takes place in terms of phonemes. The fact that vowel height discrimination is equally categorical in this central part of the vowel space and outside of it means that perception of F1 takes place in term of classificatory features, not phonemes.

Chládková's (2014) findings can be accounted for by the current model if

phonetic factors are taken into account (see below). If features have a phonetic definition, then a vowel discrimination task should have access to features based on pure acoustics. The model in **Figure 1** simply indicates that features are induced based on how segments group together, but does not limit what definition they receive once they are induced. Of course, the validity of this result should be verified once phonetic factors have been integrated into the learning model.

An ideal version of the current model should take phonetic factors into account. Even though the lack of substantive or phonetic factors in the current version of the model present in this chapter is reminiscent of Substance-free Phonology (Morén 2006), this lack of phonetic factors is merely a simplification (see below). It has been demonstrated experimentally (see, for instance, Cristia et al 2013 for results from perception experiments with infants; see also the evidence reviewed by Moreton and Pater 2012) that phonetic factors play an important role in the mental grouping of segments. Including phonetic information into the model is also essential for it to generalize to novel segments (see section 2.5). Therefore, evidence for classificatory features should be taken from both the phonetic and the structural realm, and classificatory features themselves should receive a definition in terms of phonetic dimensions.

However, in order to focus the model on the idea of feature induction for the sake of summarizing and grouping segments, I concentrated on the structural factors – since inducing phonological features without motivation from the grammar can be done on the basis of phonetics alone (see, for instance, Lin and Mielke 2008), but inducing phonological features as motivated by grammar learning can only be done with structural

factors present.

In the current emergentist model, phonological categories are learned jointly with the grammar. This can be contrasted with a traditional, innate model of phonological representation: the categories of segmental representation are innate classificatory features. The most logical learning scenario for this model would be one in which the mapping from acoustics to features is learned, and separately from that, grammatical statements are learned from the encoding of acoustic tokens into features.

Figure 2 and Figure 3 below contrast the learning scenarios implied by the standard approach with innate features, and the one followed by the current emergent model, respectively. Each arrow stands for the relation “X influences the shape of Y”. In the innate categories model, the data and the features are given, and then both are used separately to build up the grammar. In the emergent categories model, by contrast, only the data are given, and features are determined both by the data and the evolving grammar; the features themselves are then used to build the grammar, creating a loop between grammar building and feature induction.

Figure 2. Innate features model

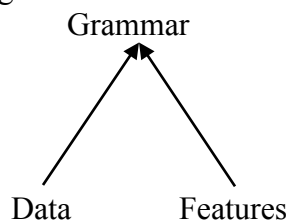
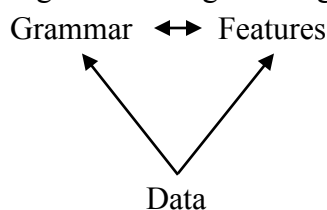


Figure 3. Emergent categories model



In this chapter, I will assume that segment units have already been learned from the acoustic signal, and my simulations only apply to the part of the learning path in Figure 3 that occurs after the data have been interpreted in terms of segments.

The precise way in which segments are learned from acoustics in this radically emergent model remains to be crystallized in future work (see, for instance, Elman and Zipser 1988, Niyogi 2004, Lin 2005, Vallabha et al. 2007, McMurray, Aslin, and Toscano 2009, Goldsmith and Xanthos 2009, Elsner et al. 2013, Elsner, Antetomaso, and Feldman 2016, Dillon, Dunbar, and Idsardi 2013, and Boersma and Chládková 2013 for proposals of how segment units are learned from acoustics). However, there is evidence that allophone or phoneme units (the distinction between allophones and phonemes will not be important for the data investigated here) are active in sound processing and phonological grammar. Experiments in speech perception and production (see, e.g., Jesse, McQueen, and Page 2007, Nielsen 2011) have demonstrated that the processing of speech makes crucial reference to segment units. For instance, Nielsen (2011) shows in an imitation study that segment categories play a role separate from both phonological features and exemplars in accounting for how listeners generalize the presence of an eccentric speech attribute (exaggerated aspiration) to new words.

Furthermore, there is evidence from phonological processes in which segment identity plays a separate role from feature identity. If segment identity must be appealed to in the grammar separately from feature identity, then grammars must use segment labels as a possible representation. One piece of evidence for segment identity comes from consonant OCP processes (McCarthy 1979, 1981a), which tend to avoid featurally similar consonants within a certain domain, but tend to allow phonemically identical

consonants in the same domain. For instance, Cochabamba Quechua (Gallagher 2014) does not allow two non-identical consonants in a disyllabic word to both be ejective, e.g., [tʃʰaka] ‘bone’, but *[tʃʰakʰa] (Gallagher 2014:338 (1a-b)). At the same time, if two consonants in a disyllabic word are identical, they may both be ejective, e.g., [tʃʰatʃʰa] ‘to soak’ (Gallagher 2014:338 (1c)).

These opposite tendencies (avoidance of similarity in non-identical pairs, non-avoidance of similarity in identical pairs) may, in principle, be explained by appealing to features only. However, this leads to a rather convoluted statement of constraints: a constraint such as OCP-[+constricted glottis] will be violated once for every two consecutive consonants which both have the feature [+constricted glottis], but only if there is at least one other feature (e.g., a place feature, as in *[tʃʰakʰa]) which the two consonants do not share. This means that a constraint specific to one feature of a consonant needs to search through all other features associated with that consonant before finding whether it is violated. A much more elegant formulation would be one where segment identity is a separate piece of information, separate from featural identity. In that case, one could say that OCP-[+constricted glottis] is violated once for every sequence of consonants which are identical in their [+constricted glottis] specification, but non-identical in their segment specification. This creates a much less computationally intensive definition of the OCP-constraint.²

Gallagher (2013) also presents evidence from an artificial language learning experiment that rules that OCP-like rules that refer to segment identity can be learned and extended to novel consonants. She shows that the data from these experiments are

² Many thanks to Joe Pater for pointing this out to me.

modeled more accurately by a modification of Hayes and Wilson's (2008) model that include reference to segment identity (Colavin, Levy, and Rose 2010) than by the same model without segment identity. This can be taken as evidence for the inclusion of segment identity into grammars.

These findings provide evidence that the phonological grammar can appeal to segments as a unit, which provides justification for assuming that segment units are somehow induced by language learning infants, and can be the basis for learning the classificatory features that are important in the grammar of a language. See also Dresher (2009, 2013, 2014) for other work that assumes that segments are the basis for learning classificatory features.

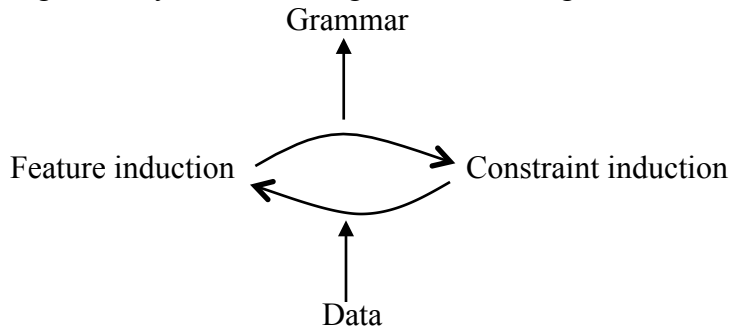
The following section will explain how the two main components of the model – constraint induction and feature induction through clustering – interact to yield maximally generalizing grammars and also abstract phonological units such as classificatory phonological features.

2.2.2 The components of the model

The radically emergent model proposed here has two main components: induction of phonological constraints, and induction of classificatory phonological features through clustering. As I will show, it is the interaction of these two components that leads to finding a grammar with maximally generalizing constraints. The induction of features opens the way for the induction of more general constraints – see, for instance, the example of [voice] in section 2.2.1. In this sense, feature induction is motivated by an external factor: the goal of maximally generalizing constraints (which will be fleshed out below).

The two components of the model interact in a cyclic way: induction of a group of constraints is followed by an attempt to find features through clustering, and the representational units found in the process of clustering are then allowed to be used in another instance of inducing constraints.

Figure 4. Cyclic model of grammar learning



Inducing a group of highly relevant constraints makes it possible to state the patterns observed in the data. For instance, if the observed tokens have no word-final [m], but there is no constraint which states that pattern, then adding a constraint against word-final [m] to the grammar will make the grammar more accurate, but adding a constraint which would require word-final [m] would not make the grammar more accurate.

Of course, it is not trivial that the non-occurrence of a certain segment in a certain position in the lexicon will lead to the induction of a constraint against it – any lexicon may contain accidental gaps. However, there is empirical evidence that at least some positional gaps in the lexicon may lead to the induction of phonotactic constraints. For instance, [ŋ] never occurs word-initially in the English lexicon, and, according to Jensen (1993:32) and (Hammond 1999), [ŋ]-initial non-words such as [ŋæpi] are ungrammatical, as confirmed by an informal survey of 8 native speakers of American English conducted by me, asking whether [ŋæpi] could potentially be an English word.

Furthermore, the statistical learning model considered here selects constraints that

maximally advance the grammar towards fitting the training data (specifically, in terms of information gain (as implemented by Wilson 2010): see section 2.3.2.1). Constraints that cover accidental gaps will typically not advance the grammar very far towards this goal, since accidental gaps are usually not systematic. On the other hand, constraints that encode systematic generalizations about the data will advance the grammar more towards fitting the training data, so that constraints that cover systematic gaps will be strongly preferred.

The model is set up so as to find a grammar in which every distinct phonological pattern is stated with the fewest number of constraints. This is what is meant by “maximally generalizing constraints” – instead of stating a phonological pattern with a large number of specific constraints (for instance, $*\#m$, $*\#n$, $*\#\eta$ for a pattern which prohibits word-initial nasals), aiming for a single constraint which triggers the pattern wherever possible (for instance, $*\#[\text{nasal}]$).

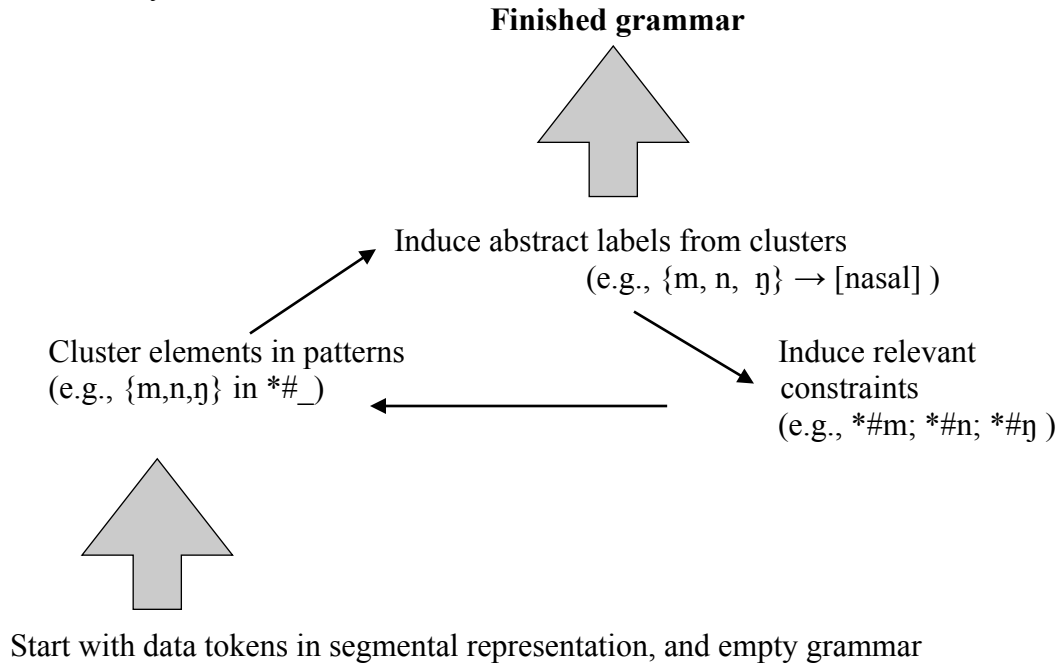
Clustering makes it possible for the learner to move toward this goal. If a language bans word-initial nasals, the constraints $*\#m$, $*\#n$ and $*\#\eta$ will be found to be highly relevant. These constraints all share the context $*\#_$ (i.e., the word-initial context), and cluster analysis will find that $\{m, n, \eta\}$ form a cluster of high relevance out of all segments of the language, when inserted in the context $*\#_$.

Clusters of representational elements thus found may be interpreted as classes of segments which undergo one and the same process. By the logic employed in Emergent Feature Theory (Mielke 2004), a previously non-existent feature label can be created and assigned to this class. In this fashion, the cluster $\{m, n, \eta\}$ will be assigned a feature label – which could be called $[\text{nasal}]$, but not that the label name is arbitrary in the model.

Once abstract labels have been induced from clustering, these labels can be used to induce new constraints.

The figure below summarizes how the model just described operates to induce grammars:

Figure 5. Summary of the model



2.2.3 Data

The data to which this model will be applied is a toy language, rooted in properties found in attested natural languages. These crucial properties have to do with the size of segment classes to which phonological patterns refer. I will show in 2.4 that it is to be expected in an emergent feature model like the one described here that the size of segment classes influences their representation in the grammar. In this subsection, I will introduce single-segment and multi-segment phonological patterns.

Phonological patterns may apply either to sets of multiple segments, or to singleton sets of segments. For instance, English epenthesizes a vowel between a stem-

final sibilant [s, z, ʃ, ʒ, tʃ, dʒ] and a following [z] (belonging to a plural or possessive suffix) – which is a pattern that applies to a set of multiple segments (at least in part of the context of the rule; Jensen 1993). This is exemplified in (5) below.

(5) Inter-strident epenthesis

$\emptyset \rightarrow [i] / [+strident] (= [s, z, ʃ, ʒ, tʃ, dʒ]) __ +/z/$

examples:

/roz/ + /z/ → roziz	“roses”
/bʊʃ/ + /z/ → bʊʃiz	“bushes”
/bætʃ/ + /z/ → bæʃiz	“batches”
but: /mov/ + /z/ → movz, *moviz	“mauves”

At the same time, English also has a clear example of a pattern that applies to a singleton set of segments. Word-initial three-consonant clusters are allowed only if the first consonant is [s], and this first consonant is followed by a stop and a liquid (Jensen 1993, Mielke 2007). By Hayes and Wilson’s (2008) convention, I will express the “only if the first consonant is [s]” clause by a negative constraint in which #CCC clusters are penalized if their first member is not [s], “not [s]” being indicated as $\wedge[s]$, as illustrated in (6).³

(6) Three-consonant clusters consist of [s] followed by a stop and a liquid

$*\# \wedge [s] CC$ ([s] = [+strident, -voice, +anterior])

examples (Jensen 1993:67):

[stre] “stray”	*[ftre], *[ntre]
[splæʃ] “splash”	*[fplæʃ], *[mplæʃ]

This pattern is by no means the only one which appeals to just one segment. An automated search of P-base (Mielke 2007) yields 13 patterns in geographically and genetically disparate languages which are encoded as applying to one single segment. A

³ The alternative to this formulation is one in terms of a positive constraint (Presley Pizzo, p.c.): reward the configuration [s] + obstruent + obstruent. Constraints that reward candidates instead of penalizing may be problematic in Standard OT (Prince 2007), but there have been proposals to use positive constraints in other frameworks. See, for instance, Kimper (2011) on positive constraints in Harmonic Serialism, and Pizzo (2013) for positive constraints in Maximum Entropy grammars.

manual search of a subset of P-base – namely, all languages whose names begin with A – yielded 11 additional patterns within that sample alone which apply to a single segment, implying that a much greater number of one-segment patterns could be found by further manual inspection⁴. From these findings, it can be established that the type of pattern in (6) is attested not only in English, but, in fact, across diverse groups of languages.

One further fact about these English data is that [s] is the intersection of two segment classes appealed to by other phonological patterns in English. One of these segment classes is the class of sibilants (as in (5)), and the other is the class of voiceless anterior coronals [t, θ, s], which are the only segments that may occur at the end of a word-final three-consonant cluster (Jensen 1993, Mielke 2007):

(7) Three-consonant clusters must end in [t, θ, s]

*CC^[+coronal, +anterior, -voice]# ([+coronal, +anterior, -voice]=[t, θ, s])

examples (Jensen 1993:69):

[nɛkst]	“next”	*[nɛksp]
[sɪksθ]	“sixth”	*[sɪksf]
[mʌmps]	“mumps”	*[mʌmpʃ]

The intersection of [s, z, ʃ, ʒ, tʃ, dʒ] and [t, θ, s] is exactly [s] – as can be seen in the diagram in (8) below.

(8) English voiceless anterior coronals in gray boxes, and sibilants in clear box

p		t		k
b		d		g
f	θ	s	ʃ	tʃ
v	ð	z	ʒ	dʒ
m		n		ŋ
w		ɹ l j		

The emergent feature model presented in 2.1-2 above induces feature labels from segment classes active in phonological patterns. This means that, if the model were to be applied to the facts summarized in (8) above, sibilants and [t, θ, s] would be assigned a

⁴ A summary of the patterns found is given in Appendix A.

feature label each (sibilants = [X], [t, θ, s] = [Y]). This, in turn, would allow the model to encode the single segment [s] in terms of these two labels ([s] = [X,Y]).

This type of re-use of existing features to describe new patterns is obligatory for single segments in this model. Since phonological features are not available *a priori*, but induced from segments' behaving as a group, one cannot invoke any features “inherent” to a single segment to describe it in terms of features (since there are no “inherent” features, but only features induced from group behavior).

Reusing existing features for new phonological constraints if they concern multiple segments can be done, too, but is not obligatory. However, there is a bias towards more general constraints through regularization (see section 2.3.2.5), which makes it more likely for the constraint selection algorithm to pick constraints that reuse existing feature labels when assessing a new pattern. This could lead to a potential feature economy effect (Clements 2003, Pater 2012) within languages.

However, generally speaking, features as induced by this procedure are not a direct basis for explanation of typological patterns, in the same sense as feature-geometrical systems such as Avery and Rice's (1989) or Clements and Hume's (1995) feature systems are. Evidently, a theory or even just a model of features must do more than simply describe the facts of a single language. However, as pointed out by Kirby and Hurford (2002) and others, typology can and should be explained by more than just innate biases.

It has been argued by, for instance, Heinz (2009) and (Staubs 2014a) that learning of grammars shapes typology. In this sense, the current model is not unlikely to make typological predictions: if an iterated learning schema is followed (Kirby and Hurford

2002), any biases in the model may be amplified over generations of speakers to yield asymmetries in the typology. In the case of the current model, there is a bias towards feature economy (Clements 2003): as shown above, the constraint selection procedure is equipped with the means (exchanging features for segments) and motivation (constraints that cover more forms improve the grammar more) to reuse previously induced features in new constraints, whereas there is a bias against inducing new features that correspond to already existing features. The fact that some phonotactic restrictions are not encoded in terms of features but in terms of segments also leads to other predictions that are made by the model if it is slightly extended (see sections 2.4 and 2.5).

Furthermore, it has been proposed by Bybee (2001) and Blevins (2004) that recurrent patterns of language use shape phonological typology. For the domain of phonological features, specifically, Mielke (2004) argues that a language use-based approach makes better predictions with the respect to the cross-linguistic typology of natural classes employed in phonological patterns. If this line of reasoning is followed, then the current model can make additional quantitative typological predictions when combined with knowledge of phonetics and language use.

These properties lay the empirical framework for answering the question asked in the introduction, namely: will the emergent feature model sketched here lead to grammars that encode even single-segment phonological patterns in terms of phonological classificatory features? Since cases like the one just sketched allow the emergent feature model to encode a single-segment pattern (only [s] word-initially in CCC) in terms of features, it is interesting to see whether the model will do this, or opt for some other type of representation. Section 2.2.4 will elaborate on this question, but

first I will describe the data which were actually offered to the model.

In order to reduce the English case just described to the bare basics – which is desirable for modeling learning, since the hypothesis space for learning both constraints and features simultaneously is quite large – I used a toy language which shared the crucial properties of the English case.

The toy language used different phonotactic restrictions compared to the English case, because the restrictions shown in (7) and (8) above require the use of a complement operator (Hayes and Wilson 2008 represent it as \wedge). This operator makes the potential space of constraints larger and the procedure of inducing phonotactic constraints more complex. For this reason, the situation found in English (the one segment referenced by phonotactic constraint A is the intersection of the segments referenced by phonotactic constraint B and the segments referenced by phonotactic constraint C) was recreated with different phonotactic restrictions.

All words in the toy language had the shape CVCVC, and the segment inventory was as in Table 1 below:

Table 1. Segment inventory of the toy language

a. consonants

p	t	k
b	d	g
m	n	ŋ

b. vowels

i	u
a	

The toy language had exactly three phonotactic restrictions, one of which is stated over one single segment (like the three-consonant onset generalization with respect to its first segment) and two of which are stated over groups of segments (like the other two

generalizations in English). These phonotactic restrictions are shown below:

(9) Phonotactic restrictions in toy language

a. single segment restriction: no word-final *m

✓panab, ✓panan, ✓panan̩ *panam

b. multiple segment restriction: no word-initial nasals [m, n, ŋ]

✓tadig, ✓badig, ✓kadig *nadig, *madig, *ŋadig

c. multiple segment restriction: no labials [p, b, m] between high vowels [u, i]

✓daban, ✓duban, ✓dabun *dubun, *dupun, *dumun, *dubin, *dibun

The single segment targeted by the restriction in (9a) can be defined by the intersection of the two groups of segments employed in the other two restrictions – labials and nasals – since [m] is the only labial nasal in the language. This is the same situation as in the English case sketched above. I will now turn to discussing how these properties of the English case and the toy language bear on the question of whether a radically emergentist model of feature learning is likely to lead to grammars which only refer to classificatory features.

2.2.4 Intuitions about predictions

In the introduction, it was stated that the standard innate model of phonological features and the emergentist model presented in sections 2.1-2.2 make different predictions about grammars with regard to the representations they use. The standard model generates grammars which only have constraints referring to features by definition (as I will show below), while it turns out that the emergent model has constraints referring to a range of levels of abstraction, including features and featurally unanalyzed segments (as will be shown in section 2.4). In this subsection, I will explain the intuitions behind these predictions.

The canonical view of phonological representation is that, in terms of segmental

representations, phonological constraints or rules refer only to bundles of (classificatory) phonological features (see, for instance, Chomsky and Halle 1968). Whether these bundles be organized in autosegmental representations (Goldsmith 1976) or not, it remains a common assumption that whenever a notation like [b] or [m] occurs, it is shorthand for an intersection of features. This view is in line with the assumption that phonological computation operates on an alphabet of universal (innate) representational elements.

If all phonological units are innate, then the best hypothesis seems to be that these innate units are phonological features (Chomsky and Halle 1968). To my knowledge, there have been no claims that non-prosodic phonological units of any other level of abstraction (for instance, atomic segment/phone units such as [b] or [i]) could be innate. Evidence in favor of the innateness of segment/phone units seems to be absent. UPSID (Maddieson and Precoda 1990, Reetz 1999) finds 919 unique segment types across 451 languages, and none of these segment types occurs in each language examined); finally, the level at which cross-linguistic generalizations can be made over segment inventories appears to be the feature rather than the individual segment (see, for instance, Clements 2003).

Thus, a view in which all phonological units are innate implies that phonological features are the only available units for phonological computation. This means that all constraints of the grammar (whether these constraints be innate or induced) should refer to these phonological features – since the translation from raw acoustic data to phonological structure proceeds by mapping acoustics onto bundles of phonological features. For this chapter, I will not run a simulation to show this – but the (strong) view

that features are the only permissible building blocks of non-prosodic phonological structure logically entails that all constraints in the grammar will refer to phonological features when dealing with segmental phenomena.

The radically emergentist view proposed here, however, does not set requirements on the units employed in the constraints in the grammars: the only goal is to build a grammar which states patterns in the most general and concise form possible. Generality can be defined along the lines of (10). This definition is similar to that used by Pinker and Prince (1988).

(10) Generality of a constraint: definition

Given a space of potential word forms Ω , constraint X is more general than constraint Y if X bans a proper superset of the forms in Ω that Y bans.

This definition is never utilized by the learner itself – instead, it relies on regularization (see section 2.3.2.5) and information gain (see section 2.3.2.1.2) to increase the generality of constraints that have non-zero weight in the grammar. The definition in (10) is somewhat *ad hoc*, and the fact that this definition does not rank constraints with a non-overlapping set of banned forms is simply because I am interested in the relative generality of constraints that encode the same pattern. However, in general, the learner prefers constraints that ban a greater set of ungrammatical forms, since these allow for representing the patterns with fewer constraints (which is preferred by regularization – see section 2.3.2.5). Constraints that ban a greater set of ungrammatical forms are likely to generalize to a great number of novel forms, and thus improve the (informally assessed) generality of the grammar.

At first sight, it appears that the goal of having maximally general constraints always converges with the goal of having maximally abstract representations (i.e.,

features) in every position of every constraint – since features help to state phonotactic patterns with a smaller number of more general constraints rather than a larger number of less general constraints. However, there are some situations in which these two goals diverge. Some patterns can be stated with maximal generality without appealing to features: this is true of patterns which refer to no more than one segment. For instance, this is true of the one-segment pattern in the toy language described in section 2.2.3, which prohibits only the single segment [m] from occurring word-finally:

(11) Single segment restriction: no word-final *m (= (9a))

✓panab, ✓panan, ✓panaŋ *panam

This pattern may be stated in terms of features only:

(12) Featural formulation of pattern

*[labial, nasal]# : One violation for every labial nasal at the end of a word.

However, the pattern can also be stated with maximal succinctness if [m] is not decomposed into phonological features:

(13) Segmental formulation of pattern

*m# : One violation for every [m] at the end of a word.

In cases of this type, then, the emergent category approach does not specify a reason why the featural formulation of the pattern (as in (12)) should be preferred over the segmental one (as in (13)). Since the featural formulation only becomes available after all the features necessary to define the one segment [m] have been induced, and the segmental formulation (*m#) is available before the induction of any of these features, it seems likely that the segmental formulation will be added to the grammar first, and the featural formulation (*[labial,nasal]#) would not be more general than the segmental formulation, and the featural formulation would therefore not be preferred over the segmental one.

This implies that one-segment patterns such as “no word-final [m]” (or [s]-initial

English clusters – see section 2.2.3) will not necessarily be stated in the grammar in terms of features, but there will be a bias toward stating such patterns in terms of individual segment categories. This is distinct from the canonical, innate feature model, in which all constraints must always refer to features.

Whereas the current model simply represents features as sets of segments, the features induced by this model can be extended to novel segments based on the phonetic properties of the segments that belong to each feature. While such an extension module was not implemented in the current learner (see section 2.5 for directions for future work), it is possible to manually approximate such phonetic extension of features learned by the model.

For instance, if the induced feature [labial] applies to all segments that share the phonetic properties that existing [labial] consonants {p,b,m} have in common, and the same is true for [nasal], then it would be a plausible guess that [labial,nasal] also applies to segments like [w̃] or [m̃], so that *[labial,nasal]# potentially bans segments other than [m]. However, the constraint *m# will never ban any segment category other than [m]. This makes the difference between *[labial,nasal]# and *m# testable.

As mentioned in the introduction, “Bach-testing” reveals the generality of a phonological pattern. Multi-segment patterns such as voicing assimilation readily spread to novel segments (as in the example given in the introduction: the loan segment [x] is recognized as voiceless, and voicing assimilation applies to the sequence /x + z/ to yield /bax + z/ → [baxs] “Bach’s/Bachs”). However, it is not clear how one-segment patterns behave on this test. In section 2.5, I will suggest ways in which Bach-testing can be applied to the output of the current algorithm.

I have reasoned above that the innate model and the emergent model vary in their predictions with respect to the representation of constraints in the grammar: the innate model leads to a grammar with features only, while the emergent model should lead to a grammar which refers to sounds in a variety of ways. However, implementing the emergent model computationally is necessary to ensure that the model does indeed make these predictions.

In the following section, I will describe a computational implementation of the emergent model described above. This implementation will perform learning of feature-like units from data expressed in terms of segments alone. As will be seen in section 2.4 afterwards, it turns out that the predictions of the emergent category model are borne out.

2.3. Computational implementation

2.3.1 General structure of the algorithm

The computational implementation of the preceding model will make use of two established machine learning techniques: modeling in a Maximum Entropy framework (Berger, Della Pietra, and Della Pietra 1996, Della Pietra et al. 1997, Goldwater and Johnson 2003, Hayes and Wilson 2008, Wilson 2010), and cluster analysis with Gaussian mixture models (Everitt et al. 2011). The novelty of the current implementation lies in the way in which these two are combined.

Maximum Entropy is a general-purpose machine-learning framework (Berger et al. 1996, Della Pietra et al. 1997) that has been applied successfully by Hayes and Wilson (2008) as well as by Wilson (2010) to the learning of phonotactic constraints from distributional patterns in a set of data tokens. See section 2.3.2.1 for a brief exposition of the mechanism of this type of learning model, and its application to phonological

learning.

However, Hayes and Wilson's model assumes that phonological features are available to the learner from the outset, while the model described in the preceding section states that features are a byproduct of building a grammar. To incorporate this latter aspect, I let the mechanism of inducing constraints interact with cluster analysis. The details of this interaction will be described in section 2.3.2.3 below. Cluster analysis will perform the function described for it in section 2.2.1: clustering of patterns in the constraint set allows for discovery of feature categories from segment categories.

The general schema of the algorithm is as follows. The procedure starts with a set of phonotactic data drawn from the toy language as sketched in section 2.2.2. The data are presented as strings of segments such as <p>, <t>, <i>:

(14) Segment inventory of the toy language

Consonants: p t k b d g m n ŋ

Vowels: i a u

The data offered to the learner are all the CVCVC forms that can be made out of the segment inventory above and which also obey the phonotactic restrictions of the toy language:

(15) Some forms offered to learner

a. offered to the learner:

panab, panan, panan, ...

tadig, badig, kadig, ...

daban, duban, dabun, ...

b. not offered to the learner:

*panam (violates "no final [m]"), ...

*nadig, *madig, *ŋadig (violate "no initial nasals"), ...

*dubun, *dumun, *dibun (violate "no labials between high vowels"), ...

At the outset, the Maximum Entropy model has no constraints, meaning that every possible representation has equal likelihood (see section 2.3.2.1.1). The set of possible

representations considered by this learner consists of all CVCVC combinations that can be made out of the segment inventory – so that this includes both the phonotactically legal forms, as exemplified in (15a), and the phonotactically illegal forms, as exemplified in (15b).

The reason why only CVCVC forms are considered is a purely practical one: these forms suffice to observe the activity of all three phonotactic constraints in the toy language. A fuller and more realistic simulation would include forms of other phonotactic shapes (such as VC, CVC, CVCV).

However, since all data offered to the learner are of the shape CVCVC, it is necessary to also restrict the hypothesis space of the grammar to CVCVC forms – otherwise, the grammar would have to account for the absence of data of the shape VC, CVC, CVCV, CVCC, etc. This is because of the nature of the Maximum Entropy learner used here: the grammar must fit the statistical distribution in the data, including gaps, with maximal accuracy (see Berger et al. 1996, Della Pietra et al. 1997, Goldwater and Johnson 2003, Hayes and Wilson 2008, Wilson 2010 for more on Maximum Entropy learners).

To get from this initial stage to the goal of having a grammar model with maximally general constraints, a loop in which constraint induction and clustering interacted is repeated until the grammar was judged as being maximally general. The loop contained five steps:

(16) Steps inside the looped part of the algorithm

1. constraint selection
2. selection of contexts for clustering
3. clustering itself
4. creation of feature labels from clusters
5. weighting of constraints selected at step 1 in the grammar model

A grammar was judged as being maximally general when the constraints currently

in the grammar, and their weights, created such a probability distribution that the grammatical CVCVC forms had at least 95% of all the likelihood. The presence of regularization (see section 2.3.2.5) in the model makes it impossible for the model to assign a large amount of likelihood to the grammatical (observed) forms if the constraints in the model are too specific. This means that a high likelihood for the grammatical forms entails that the constraints have a certain level of generality.

Section 2.3.2 will now give descriptions of each of the five steps in the loop summarized in (16). The full code of the algorithm can be found in an additional file uploaded with this dissertation in ScholarWorks@UMassAmherst.

2.3.2 The steps of the algorithm

2.3.2.1 Selection of constraints

The first step of the iterated part of the algorithm (see (16) above) is the selection of a small group of constraints for the stochastic grammar. The intuition behind the selection procedure is to find one constraint or a small group of closely related constraints which correspond to a pattern in the data. In practice, this was done with a hill-climbing algorithm that made use of a rudimentary form of evolutionary algorithms (Ashlock 2006) – to find a local peak of information gain (see section 2.3.2.1.2) relative to the current grammar. This hill-climbing algorithm was used so that only the most efficient constraints were selected – which was necessary since there was no explicit, substantive mechanism that tried to extract patterns from comparison of grammatical or ungrammatical patterns. Rather, a constraint’s potential improvement of the grammar’s fit to the data was the only way to gauge that constraint’s appropriateness to the data set.

In order to explain constraint selection in more detail, I will first briefly introduce

Maximum Entropy grammars, and then introduce information gain. After this, I will explain the mechanism of constraint selection in more detail.

2.3.2.1.1 Maximum Entropy learning for phonotactics

MaxEnt models (Berger et al. 1996, Della Pietra et al. 1997, Goldwater and Johnson 2003, Hayes and Wilson 2008, Wilson 2010) make use of constraints (for instance, OT-style phonological constraints)⁵ to generate a probability distribution over objects/events (for instance, phonological input/output mappings or phonological output forms). The distinctive characteristic of the MaxEnt model is that every constraint is assigned a weight such that overall information-theoretic entropy is maximized (i.e., the model makes minimal assumptions about unknown objects/events to determine the weights of constraints).

Because this type of model weights constraints based on the training data given to the model (as well as a regularization term) and does not require explicit negative evidence, MaxEnt models are very useful for modeling the learning of phonological grammars from positive data. Since language-acquiring infants learn their grammars from positive data (Brown and Hanlon 1970, Marcus 1993), these models can give us insight into the acquisition of phonological grammar (Hayes and Wilson 2008).

MaxEnt models can be seen as a probabilistic version of Harmonic Grammar (Pater 2009, Potts et al. 2010): both analytic frameworks share the property of having numerically violable constraints which have weights instead of ranks. In contrast to non-probabilistic versions of Harmonic Grammar, however, MaxEnt models do not appoint a

⁵ In the field of Natural Language Processing, the statements which phonologists call “constraints” are instead called “features” (see, e.g., Della Pietra et al. 1997). I will follow the phonologists’ usage.

winning candidate out of a list of candidate outputs, but, instead, they define a probability distribution over output candidates.

Since learning in this case will be purely phonotactic (following Hayes and Wilson 2008), the grammar defines a distribution over all possible output forms, as if they all come from the same input. As had already been mentioned in section 2.3.1, the toy language only has “words” of the shape CVCVC, so that the set of possible output forms was also built on that pattern (see (15)). For modeling a more realistic language that has a variety of word shapes, a more varied set of output forms would have to be considered.

Each of the possible output forms is assigned some probability by the grammar. Even though each form will have a very small probability (the number of possible forms is 6,561), phonotactically better candidates will still have a much higher share of probability than phonotactically worse candidates.

As had been said above, the goal of a MaxEnt model is to maximize entropy in the system. Entropy is maximized when the probability distribution assigned to candidates by the grammar maximally approaches the distribution of data points input to the learner (Manning and Schütze 1999). For this reason, the weights of the constraints in the model are adjusted so as to minimize the discrepancy between predicted probabilities (q) and observed probabilities (p) of data points, which are defined as follows:

(17) Observed probability of a candidate: count of observations for candidate x divided by total number of observed tokens of any candidate (Ω stands for the set of all candidates)

$$p(x) = \frac{|x|}{\sum_{y \in \Omega} |y|}$$

(18) Predicted probability of a candidate given a MaxEnt model

$$q(x) = \frac{e^{H(x)}}{\sum_{y \in \Omega} e^{H(y)}}$$

where
 $H(x) = \sum_i w_i \times C_i(x)$,
 $C_i(x)$ is the penalty constraint C_i assigns to candidate x , and
 w_i is the weight assigned to C_i

The discrepancy between these two distributions is obtained by finding the Kullback-Leibler (K-L) divergence (Kullback and Leibler 1951) of q from p . K-L divergence (also called relative entropy) is defined as in (19). Since K-L divergence is calculated with the natural logarithm in this case, instead of the base 2 logarithm, its units of measurement are nats instead of bits.

(19) Kullback-Leibler divergence of model distribution q from sample distribution p

$$D_{KL}(p \parallel q) = \sum p(x) \times \log \frac{p(x)}{q(x)}$$

To maximize entropy, the weights of constraints are adjusted so as to minimize the K-L divergence of the model distribution, q , from the sample distribution, p :

(20) Objective function of MaxEnt grammar (without regularization term)

$$Obj = \min_w D_{KL}(p \parallel q) = \min_w \sum p(x) \times \log \frac{p(x)}{q(x)}$$

As had been mentioned in section 2.3.1, the model also had a component in it that encourages generalization: this was an L2 regularization term, which biases the grammar towards having low cumulative constraint weights (i.e., having the lowest possible weight for every constraint, even if this is a zero weight). Specifically, the difference between every constraint weight in the model and a model mean is squared, and the results of this are added together and divided by a constant (twice the variance).

(21) Objective function of MaxEnt grammar with regularization term

$$Obj = \min_w \left[\sum p(x) \times \log \frac{p(x)}{q(x)} - \sum_i \frac{(w_i - \mu)^2}{2\sigma} \right]$$

In the current model, this function was minimized by the L-BFGS-B method built in into R (R Core Team 2013) – see <http://stat.ethz.ch/R-manual/R-patched/library/stats/html/optim.html> for documentation and references. In principle, any minimization method may be used.

The regularization term was given a mean of 0 (so that constraints would be biased toward have as low a weight as possible, which is standard practice), and a very low variance of 1,000. This low variance means that there will be a very strong pressure for the grammar to have maximally general constraints, which will maximally bias the grammar towards having features. This means that it will be surprising if the grammar does not refer to features in some cases.

Since the objective function must be minimized, the optimum of the function strikes a compromise between maximum entropy and minimum squared summed constraint weights. This has the effect that, when a pattern is represented by many constraints that each only represent a subset of the forms banned by the patterns (e.g., **#m*, **#n*, **#ŋ* for a pattern that bans all nasals word-initially), these constraints will not be allowed to have high enough weight to truly distinguish between grammatical and ungrammatical candidates, while assigning weight only to a single constraint for the same pattern (e.g., **#[nasal]*) be punished much less, so that this single constraint will be allowed to have a higher weight. The constraints **#m#n*, **#ŋ* each need the same weight as **#[nasal]* to account for the data, but if **#m*, **#n*, **#ŋ* are actually given these weights, their penalty is three times the penalty for **#[nasal]*; to mitigate this penalty, the model is forced to give each of **#m*, **#n*, **#ŋ* a lower weight than they need to fully account for

the pattern. In fact, as will be seen in section 2.1.3.2.5, $\#m$, $\#n$, $\#n$ will be given zero weight because of the strength of the regularization term used here.

As we will see, this means that, in order to reach the criterion of 95% total model-assigned likelihood (q) on grammatical candidates, mentioned in section 2.3.1, the general constraints need to be given sufficiently high weight. Analyses in which general constraints have not been found and analyses whose high-weighted constraints are insufficiently general will have all their constraint weights forced down by the regularization term, so that the weights of individual constraints will be so low that the grammar will not be able to ban ungrammatical forms. I will return to the effects of the L2 prior in section 2.3.2.5, where I will discuss its effects on the way in which constraint weights are updated in the course of the simulation.

2.3.2.1.2 Information gain

The concept that will be crucial to constraint selection is information gain (see Della Pietra et al. 1997, Wilson 2010). This is an information-theoretic measure that estimates how much a new constraint could maximally improve the grammar model's fit to the training data. This measure was chosen so that, out of all potential constraints, only those that best improve the grammar's fit to the training data could be added to the model. Since there is no mechanism that extracts patterns from the raw data, this is one possible way to ensure that constraints that correspond to the patterns in the data are chosen.

Formally speaking, information gain measures the maximal drop in the divergence of the predicted distribution (q) from the observed distribution (p) that could happen if the new constraint C^* were added to the grammar with weight w^* . The weight

of the new constraint may be varied for the purpose of maximization, but the weights of the other constraints in the grammar are kept constant. This latter restriction is put in place simply for computational convenience – it would be even more informative to optimize the weights of all existing constraints together, but this would take up more computational resources. Formally, information gain is expressed by the following formula (where $q_{w^*C^*}$ stands for “the distribution over candidates defined by the current grammar q to which constraint C^* is added with weight w^* ”):

(22) Information gain

$$G(w^*, C^*) = \arg \max_{w^*} [D_{KL}(p \parallel q) - D_{KL}(p \parallel q_{w^*C^*})]$$

Since information gain is based on K-L divergence, and K-L divergence is measured in nats, information gain is also measured in nats. I will use this measure in the procedure of constraint selection, which will be described in the next subsection.

2.3.2.1.3 Constraint selection

The constraints used in this implementation were negative (penalty-assigning) constraints that penalized sequences of two or three elements (i.e., bigrams or trigrams). These elements could be either a word boundary or members of the set of segmental representational elements, σ' . The set σ' consists of all the segments in the segment inventory of the toy language, and all the phonological features currently in the model⁶.

The following are examples of possible constraints:

(23) Possible constraints

*t#

*a[labial]

*amu

⁶ As will be seen in section 2.3.4, features were actually implemented as sets of segments – which were inserted in constraint definitions as alternative sets (“m OR n OR ŋ”, encoded as [mnŋ] in regular expressions).

The penalty (number of violations) assigned to a candidate by a constraint was computed by encoding the sequence of elements in the constraint definition as a regular expression, and counting the number of occurrences of the expression in the candidate.⁷

Constraints were induced through a procedure driven by the concept of information gain as laid out above. The process of inducing a constraint (or small group of constraints) consisted of two steps: a random search step, and an optimization step, each of which will be detailed below.

The random search step found a ‘seed’ constraint that was at least marginally relevant to the data for the optimization (hill-climbing) step using rejection sampling (criterion: information gain is at least 0.01 nats), to minimize the number of calculations to be performed by the optimization step. The optimization step then gradually modified this ‘seed’ constraint to find a peak in information gain, i.e., to find a constraint that was, among similar constraints, the most useful to add to the grammar. This optimization step was also used for practical purposes, because the search space of constraints is too large (2,184 at the outset, increasing with every feature induced) to reach high information values (of 0.05 or higher) with the use of rejection sampling within reasonable time.

At the random search step, the algorithm generated a random bigram or trigram built out of the elements of σ' (plus #, the word boundary) and computed that constraint's information gain value. This was repeated until a constraint with an information gain of at least 0.01 nats was found. For instance, this could have been the constraint `*#ba`.

However, in order to add only the most useful constraints to the grammar, this

⁷ The match of a constraint's structural description was evaluated ‘greedily’, i.e., for all sequences of 2 or 3 sounds in a word, even if these overlap. For instance, a constraint `*bb` would be violated twice in a word form [abbba]

constraint with an information gain of at least 0.01 nats was subjected to further optimization. The optimization step continuously modified its input constraint until a peak of information gain was reached. Three types of modification were used:

(24) Types of modification

1. deletion of a random element (for trigrams)
or insertion of a random element in a random position (for bigrams)
2. change a random segment to another segment (not applicable to features)
3. change a random feature or segment to another feature⁸

Every one of these modifications produced a “neighboring” constraint, in the sense that the output of modification is one step away from the input of modification. Whenever a modification resulted in a higher information gain, the resulting neighboring constraint was kept, and this neighboring constraint itself was used as the basis for a new series of modifications. Whenever a modification did not result in a higher information gain, the neighboring constraint created was discarded. This is a technique loosely based on evolutionary algorithms (Ashlock 2006).

The modifications were halted as soon as 10 modifications in a row yielded no improvement in terms of information gain: in this case, it is assumed that a peak in information gain value has been found. One possible path of improvement is the following:

(25) Example of improvement path

*#ba → *#ma → *#m

The output of the entire optimization step was the constraint from which no improving modifications could be made, plus whichever neighboring constraints could be constructed that had the same information gain (within some range). For instance, for

⁸ If only one feature is available at the time when this modification is performed, then either a segment is replaced by that single feature, or that feature is replaced by a random segment.

*#m, these neighboring constraints with the same information gain were *#n and *#ŋ. The output of the optimization step for this example, therefore, is the set { *#m, *#n, *#ŋ }.

If there were no neighboring constraints with the same information gain value, only the constraint which represented the peak in information gain was output by the procedure (for instance, *m# had no neighboring constraints with similar information gain values). The one or several constraints selected by this two-step algorithm were then passed on to the second step – which was the creation of contexts in which feature induction was to take place, based on the constraints just induced.

2.3.2.2 Context creation

Once a group of constraints had been selected at the first step – for instance, the set { *#m, *#n, *#ŋ } – these constraints became the basic material for feature induction. Feature induction was done by clustering segments by their information value when inserted into a constraint context. The first step in this process was to find all possible contexts inherent in the constraints just selected.

A context was defined as a phonological configuration with one time slot missing. As a consequence of this definition, the set of contexts that could be created from a single constraint consisted of every way of removing one time slot from that constraint. For instance, the single constraint *ubi would have yielded the following set of contexts:

(26) Set of contexts for *ubi

*_bi
*u_i
*ub_

This procedure of finding contexts was repeated for every constraint that was in the set selected at step 1 – and then all unique (non-repeating) contexts thus found were retained

and passed on to the next step. For the set of constraints $\{*\#m, *\#n, *\#\eta\}$, which had been selected at step 1, this yields the following contexts:

- (27) Set of contexts for $\{*\#m, *\#n, *\#\eta\}$
 $*_m$ (found for $*\#m$)
 $*_n$ (found for $*\#n$)
 $*_ \eta$ (found for $*\#\eta$)
 $*\#_$ (found for all three constraints)

The contexts thus found were passed to the next step of the loop, which was clustering – a crucial step toward finding features.

2.3.2.3 Clustering

Once contexts were found, as in the preceding step, these contexts were tabulated against the segment inventory of the toy language. This was done so that it would be possible to identify which segments are more active in a certain context. The table constructed for the contexts obtained at the previous step ($*_m, *_n, *_ \eta, *\#_$) is shown below:

Table 2. Context-and-filler table for $\{*\#m, *\#n, *\#\eta\}$: all possible contexts are on the vertical dimension, all possible fillers are on the horizontal dimension

	a	i	u	p	t	k	b	d	g	m	n	η
$*_m$												
$*_n$												
$*_ \eta$												
$*\#_$												

This table was populated by the information gain values (see section 2.3.2.1.2 above) for the constraints constructed by inserting the segment corresponding to the column in the context corresponding to the row. For instance, the cell in the “u” column and the “ $*_m$ ” row stands for the constraint $*um$.

The table below shows information gain values for the $*\#_$ row of the previous table:

Table 3. Context-and-filler table values for *#

	a	i	u	p	t	k	b	d	g	m	n	ŋ
*#	0.001	0.001	0.001	0.002	0.002	0.002	0.002	0.002	0.002	0.015	0.015	0.015

Every row of the table, filled out according to this procedure, was then passed on to the clustering procedure. Clustering was done by fitting a Mixture of Gaussians model to the data (which is a special case of the Finite Mixture model – see Everitt et al. 2011). A Mixture of Gaussians model captures multimodal distributions by positing that the data come from several populations, with every population being represented by a separate Gaussian distribution with its own mean and standard deviation. The overall likelihood of a data point is a weighted sum of its likelihood under every Gaussian in the model. The model also calculates a probability that a data point belongs to one of the populations based on that data point’s relative likelihood under each Gaussian. Because of this, if a 2-component Mixture of Gaussians model is fit to a data set, its data points can be clustered into 2 populations by assigning each data point to whichever population it has at least a 0.5 probability of belonging to.

For every context (as represented by a row in the table), an equal-variance Mixture of Gaussians model was fit to the vector of information gain values corresponding to that context⁹. The model was always given two components (two Gaussians) to work with, because the desired outcome was a separation of the segment set into two categories: the segments that are active (banned by a constraint) in a certain context, and the segments that are inactive (not banned by a constraint) in a context.

The information gain values in every cell of a row of the table represent the

⁹ Because the vector of values to cluster over was so short, and the information gain values within a cluster tended to be identical, the model was unable to fit combinations of statistical distributions to these data. This was remedied by adding random noise to the data (an amount of 0.0000001 was added to random table cells).

degree to which inserting a segment in a constraint context and adding it to the grammar improves the grammar's fit to the data. For a given constraint context (e.g., *#_) and a given segment (e.g., [b]), this measure provides an estimate of the likelihood of whether that segment participates in the pattern denoted by that context (“How likely is it that [b] participates in the pattern [no word-initial _] ?”).

The participation of a segment in a phonological pattern reveals its phonological function. It is in this sense that a clustering model over information gain values in a context can reveal something about the phonological function of groups of segments.

As can be seen above visually, the values for [m], [n], and [ŋ] are much higher than for the other segments within the context *#_ (which is a reflection of the fact that the data lack precisely [m n ŋ] word-initially). The 2-component Mixture of Gaussians model found exactly that division:

Table 4. Division into higher-mean (bolded) and lower-mean component (not bolded)

	a	i	u	p	t	k	b	d	g	m	n	ŋ
*#_	0.001	0.001	0.001	0.002	0.002	0.002	0.002	0.002	0.002	0.015	0.015	0.015

After the Mixture of Gaussians model had been fit, the component with the highest mean (corresponding to the “active” segments) was used as the basis for a new feature, when appropriate (see the next section on what was seen as “appropriate”). The clustering model was run on every context in the table, so that every context discovered at the preceding step (in our example, this would be *_m, *_n, *_ŋ, *#_) could potentially give rise to a new feature.

2.3.2.4 Feature induction

Feature induction was based on the following principle: if the Gaussian with the higher mean defined (using the likelihood of each data point given that Gaussian) a class

of more than one segment, that multi-segment class was a candidate for inducing a new feature. Features that were homonymous with a single segment were disallowed to ensure that features abstracted over segments in a non-vacuous way – see below for more discussion.

Once a Mixture of Gaussians model was fit to the data for one of the contexts, the Gaussian component with the higher mean was taken (since this Gaussian component stands for the segments active with that context), and the likelihood of every segment under that Gaussian component was computed. An approximation of the likelihood vector for the context $*\#_$ is displayed below:

Table 5. Likelihood vector for higher mean Gaussian in context $*\#_$

	a	i	u	p	t	k	b	d	g	m	n	ŋ
$*\#_$	5.2×10^{-9}	5.2×10^{-9}	5.2×10^{-9}	5.2×10^{-9}	5.2×10^{-9}	5.2×10^{-9}	5.2×10^{-9}	5.2×10^{-9}	5.2×10^{-9}	1	0.99	0.99

This likelihood vector was used for deciding whether a certain segment was part of the class of segments active in a context. As is standard practice in Finite Mixture models, a segment was classified as part of the active class whenever the likelihood of that segment was at or above 0.5. In the example above, this yields [m, n, ŋ] as being part of the class active in the context $*\#_$.

There were exactly two situations in which such a likelihood vector was not stored. One situation was when a Mixture of Gaussians model of the desired kind could simply not be fit to the information gain values of a context, for instance because there was too little variation between information gain values in that context¹⁰.

The other situation is when the model gave only one segment ≥ 0.5 likelihood

¹⁰ I used the Mclust implementation of Mixture of Gaussian clustering from the mclust package, version 4.2, in R (Fraley, Raftery, and Scrucca 2012). When this implementation failed to fit a model with 2 equal variance components to the data for a context, no likelihood vector could be recorded.

under the higher mean Gaussian – in other words, when the model threatened to induce a feature that is assigned to one single segment only¹¹.

Such was the case for the context *₋#, for instance: only [m] was banned word-finally, so that only the segment [m] would be predicted by the model to be one of the active segments in that context. Since intersections of features were computed and allowed in constraints, and the intersection of [labial] and [nasal] yielded exactly the segment [m] in the toy language, the grammar still has a way of appealing to the single segment [m] through features alone.

The reason why feature labels that stand for one segment were not allowed is to let feature labels always be more general than their corresponding segments. If the generality of a representational unit is measured in terms of how broad a part of the articulatory and acoustic spectrum it covers, then this means that feature labels must cover more articulatory or acoustic ground. Since the classificatory features in my (simplified) model are defined solely in terms of the segments they classify, a feature that classifies only one segment of a language denotes only that segment, and nothing more. For example, a feature [X] which stands for the segment [m] and no other segments would not be more general than [m], since the set of acoustic realizations of [m] would be the same as the set of acoustic realizations of [X].

Crucially, however, the absence of features that refer to one segment does not bar

¹¹ While it is advantageous in hierarchical feature assignment models such as the one proposed by Dresher (2009) to assign a feature to a single segment (because the function of features is to contrast segments), it is not advantageous to do so in the current model, because the function of features in this model is to summarize phonological behavior and make phonological patterns more easily representable. Despite this, it of course remains essential to account for the notion of phonological contrast in any model, and in future work the model may be extended to account for contrast as well.

the system from referring to a single segment by feature labels. As I will explain toward the end of this subsection, features were allowed to combine to define a set of segments. For instance, the combination of [labial] ($=\{p,b,m\}$) and [nasal] ($=\{m,n,\eta\}$) yields [labial, nasal], which stands for [m]. It is in this way that the model was able to refer to single-segment classes through features – and, as will be seen in the results section, 2 of the 32 runs of the simulation actually did have a constraint *[labial,nasal]#. This means that the result that grammars obtained from my learning model generally did not refer to single-segment classes through intersections of features is not pre-encoded in the model itself, but truly is an emergent fact.¹²

The assignment of feature labels to segments proceeded through the likelihood vectors that were stored for each Mixture of Gaussians model (and, thus, for each context). A feature label was assigned to the segments that had at least 0.5 likelihood under that model.

To prevent the same segment class from being assigned several feature labels (since the same class can be active across multiple contexts; for instance, the set [m,n,η] could be active in several contexts, but it should not be assigned 4 different labels), the following safeguard was used. Whenever a new likelihood vector had been found through Mixture of Gaussians analysis of some context, the similarity of that new likelihood vector to every previously stored likelihood vector (if any were present at that point) was computed. The similarity between two likelihood vectors was computed based on the “profile” of the two vectors: are relatively larger values in the same place in both

¹² Of course, the fact that the grammars never referred to single-segment classes through single features, like the example of a potential feature [X] standing for just [m], is pre-encoded.

vectors¹³? If the similarity to some existing likelihood vector exceeded a certain threshold (0.9 on a scale of 1 in this case), then the new likelihood vector was stored under the label of that existing likelihood vector. Otherwise, the new likelihood vector was stored under a new, randomly generated label.

One alternative to this similarity metric would be to simply delete vectors that lead to the same sets of segments (i.e., that have the same set of segments with a likelihood of at least 0.5). However, since information gain is merely an estimate of a constraint's usefulness in the analysis, sometimes very similar but non-identical likelihood vectors are induced. A similarity metric makes it possible to gauge whether such pairs of vectors are likely to stand for the same pattern: high similarity (≥ 0.9) means that it will be assumed that both vectors represent the same pattern, whereas low similarity means that they will be presumed to instantiate different patterns.

For instance, at some iteration of the algorithm after the vector in Table 5 had been stored, likelihood vectors like the ones in Table 6 and Table 7 may be found. While the vector in Table 5 above lifts out the segments [m,n,η], the vectors in Table 6 and Table 7 below lift out the segments [n,η] only.

Table 6. A likelihood vector non-identical to the one in Table 5, similarity < 0.9

	a	i	u	p	t	k	b	d	g	m	n	η
*_u	5.2 x 10 ⁻⁹	5.2 x 10 ⁻⁹	5.2 x 10 ⁻⁹	5.2 x 10 ⁻⁹	5.2 x 10 ⁻⁹	5.2 x 10 ⁻⁹	5.2 x 10 ⁻⁹	5.2 x 10 ⁻⁹	5.2 x 10 ⁻⁹	0.45	0.99	0.99

¹³ Similarity was calculated by normalizing both vectors with an L2 normalizer so that they both summed to 1, and then taking the dot product of these vectors: $Similarity(v, w) = \frac{[v_1, \dots, v_n]}{\sqrt{v_1^2 + \dots + v_n^2}} \cdot \frac{[w_1, \dots, w_n]}{\sqrt{w_1^2 + \dots + w_n^2}}$; the large numbers in these vectors will be closer to 1, and the smaller numbers will be closer to 0. The only pairings that bring the dot product close to 1 are pairings of two large numbers.

Table 7. A likelihood vector non-identical to the one in Table 5, similarity > 0.9

	a	i	u	p	t	k	b	d	g	m	n	ŋ
*a ₋	5.2 x 10 ⁻⁹	5.2 x 10 ⁻⁹	5.2 x 10 ⁻⁹	5.2 x 10 ⁻⁹	5.2 x 10 ⁻⁹	5.2 x 10 ⁻⁹	5.2 x 10 ⁻⁹	5.2 x 10 ⁻⁹	5.2 x 10 ⁻⁹	0	0.99	0.99

The vector in Table 6 has a similarity metric of 0.95 with respect to Table 5, whereas the vector in Table 7 has a similarity metric of 0.81 with respect to Table 5. This means that the vector in Table 6, in which [m] is almost at the threshold of being considered a part of the segment class, will be considered to be overlapping with the one in Table 5. Likelihood vector for higher mean Gaussian in context *#₋ and will be deleted. At the same time, the vector in Table 7, in which [m] is categorically outside the segment class, will be considered separate from Table 5, and will be stored.

Thus, to summarize, a new label was generated every time a fit of the Mixture of Gaussians model found a likelihood vector over the set of segments which was unlike any previously stored vector. For each labels which was found at a given iteration of the cycle, its likelihood vector was converted into a set of segments by taking all the segments that had at least 0.5 likelihood under that label (the same criterion as described above)¹⁴. For instance, if the likelihood vector in Table 5 above was assigned the random label “XM”, then the procedure just described yields a statement like “XM = {m,n,ŋ}”.¹⁵

Even though labels assigned to clusters of segments were arbitrary, I will use the names [labial] and [nasal] for {p,b,m} and {m,n,ŋ}, respectively, when presenting the results of the simulations in section 2.4. However, the use of the labels [labial] and [nasal] does not mean that the classes that they denote have any meaningful phonetic

¹⁴ For labels that had more than one likelihood vector stored under them, all these vectors were averaged and the segments which had at least 0.5 likelihood under that averaged likelihood vector were said to belong to that label.

¹⁵ Each label was a two-letter code to ensure that a plentiful supply of labels was available.

interpretation.

Since single segments can only be referred to by a combination of classificatory features and not by single features (given the definition of features given above; for instance, [m] can only be referred to by [labial, nasal], not by any single feature), it was necessary to also define segment sets which correspond to a combination of features.

After the segment sets corresponding to the new labels induced at an iteration had been determined (for instance, “ $XM = \{m, n, \eta\}$ ”), the following procedure was followed: for each newly induced label, the intersection of that label's segments and each other label's segments was computed. For instance, if the set of previously induced labels was as in (28) below, the intersection between the newly induced label “ $XM = \{m, n, \eta\}$ ” and these labels would be as in (29).

(28) Examples of arbitrary labels induced for sound classes

LO = {p, b, m}

KV = {a, i, u}

TR = {i, u}

(29) Intersections between $XM = \{m, n, \eta\}$ and the features in (28)

{m}

{}

{}

The sets of segments corresponding to new single labels and new combinations of labels were subsequently added to σ' , the repertoire of representational units out of which constraints could be built (see section 2.3.2.1.3). In this fashion, the new features were allowed into constraints that were to be induced at the next iteration of the cycle.

2.3.2.5 Constraint weighting

Once the constraints selected at the first step had been exploited for feature induction at steps 2 to 4, these originally selected constraints were added to the grammar.

No additional constraints were induced during the feature induction stage – the features induced at that time were made available for feature induction at the next iteration. This setup was followed so that the feature-based constraints that are indeed induced at the next iteration be peaks of information gain among similar constraints – whereas the feature induction procedure did not have a way of ensuring this.

The newly induced constraints were added to the previously induced constraints (if any), and the weights of new and old constraints alike were then adjusted to minimize the objective function (see (21) in section 2.3.2.1.1).

The newly induced constraints started out with a weight of zero, while the old constraints started out at their previous weight; these initial weights were then optimized to minimize the objective function. The optimization procedure used here, a pseudo-Newtonian update procedure (see the references in footnote 6 in section 2.3.2.1.1), is sensitive to the initial weights given to it, since the optimum is found by gradually changing constraint weights using a approximation of the second-order gradient of the objective function.

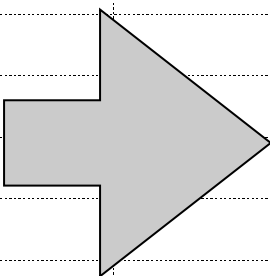
For instance, if the constraints $*\#m$, $*\#n$, $*\#\eta$ were added to an empty grammar, the constraint vector and the weight vector before and after weight optimization would look as displayed in Table 8 below. The weight of 6 happens to be ideal for each of these constraints to express the absence of word-initial [m], [n], and [η]. The constraints end up with equal weights since they punish the same number of candidates, because the number of candidates starting in [m] is the same as that of candidates starting in [n], or [η].

Table 8. Grammar with constraints $*\#m$, $*\#n$, $*\#\eta$

Constraint	$*\#m$	$*\#n$	$*\#\eta$
Weight	6	6	6

However, if the more general constraint $\#[\text{nasal}]$ was added at the next iteration, a “zero reset effect” took place: because $\#[\text{nasal}]$ completely overlaps with the functions of $\#m$, $\#n$, and $\#\eta$, and regularization prefers a lower sum of constraint weights, it is more advantageous to assign more weight to $\#[\text{nasal}]$, and less weight to $\#m$, $\#n$, and $\#\eta$. The fact that $\#m$, $\#n$, and $\#\eta$ are reset to zero, and not to some other number lower than 6, follows from the fact that regularization was very strong: the variance in the prior was set very low¹⁶ (variance = 1,000), so that there was a strong bias against assigning any amount of weight to constraints that were not general enough. In Table 9 below, the left hand side represents the grammar with addition of $\#[\text{nasal}]$ before optimization, whereas the right hand side represents the same grammar after optimization.

Table 9. Reset of specific constraints to zero

Before optimization			After optimization	
Constraint	Weight		Constraint	Weight
$\#m$	6		$\#m$	0
$\#n$	6		$\#n$	0
$\#\eta$	6		$\#\eta$	0
$\#[\text{nasal}]$	0		$\#[\text{nasal}]$	8

Had the variance parameter in the regularization term been set to a higher value, which means that the drive towards low constraint weights would be smaller, then less

¹⁶ Simulations run with a weaker bias (for instance, with variance = 5,000) do not exhibit a zero-reset effect, but these simulations also do not find more general constraints which supersede previously induced specific constraints (as is the case for $\#[\text{nasal}]$ versus $\#m$, $\#n$, $\#\eta$). The latter might be avoided if the selection criterion for constraints – information gain – were to be replaced by optimizing weights for all constraints during the constraint induction procedure. However, this would be computationally costly, since information gain is computed many times both during the constraint induction procedure and during the feature induction procedure.

general constraints such as $*\#m$, $*\#n$, $*\#\eta$ would actually be active in the grammar – which would have supported the hypothesis set out in the chapter without effort: constraints that refer to segments are in the grammar alongside constraints that refer to features.

On the other hand, the zero-reset effect that is a function of the strong regularization term is perfectly in line with the standard hypothesis that phonological grammar refers to features only: since the constraints that refer to the individual segments $[m]$, $[n]$, and $[\eta]$ have zero weight in the right-hand part of Table 9, the grammar no longer refers to the individual segments $[m]$, $[n]$, $[\eta]$ as far as banning them in word-initial position is concerned.

In this manner, the zero-reset effect gives the standard hypothesis of an all-feature grammar a head start: if the zero-reset effect happened for all instances where a feature-based constraint replaces a segment-based constraint, the learner would end up with an all-feature grammar.

However, the zero-reset effect, as described above, only happens when a new constraint is more general and covers a strict superset of the forms that one or more old constraints account for. When a new constraint is homonymous with an old constraint, both constraints receive non-zero weight after optimization. For instance, when a grammar has the constraint $*m\#$, and the constraint $*[labial,nasal]\#$ is added to that grammar, both constraints retain non-zero weight after optimization, so that the grammar refers to both features and segments.

This difference between the two situations just described (feature-based constraint punishes a strict superset of forms punished by segment-based constraints; feature-based

constraint is homonymous with segment-based constraint) follows from the nature of the L2 prior. This is because a new constraint like *[labial,nasal] does not take over the function of more than one old constraint – so that assigning all weight to that constraint cannot lessen the penalty imposed on high weights for individual constraints (since adding weight to *[labial,nasal] does not take away weight from more than one constraint).

After constraints were weighted, the total likelihood assigned by the resulting model to the grammatical candidates was assessed. If this likelihood was less than 95%, another iteration of the cycle described in this section (section 2.3.2) was initiated. Otherwise, the constraints and weights as fixed at the latest iteration were output as the final grammar.

2.4 Results

A run of the computational simulation as described in the previous section yields a grammar in the form of a set of constraints, and a weight for each constraint, as illustrated for an example run in Table 10. As explained in section 2.3.2.1.1, these constraints and their weights generate a probability distribution over potential CVCVC forms.

Table 10. Results of example run

Constraint	Weight
*m#	2.45
*ubi	0
*upi	0
*umi	0
*umu	0
*imu	0
*imi	0
*ipi	0
*ipu	0
*upu	0
*ubu	0
*ibu	0
*ibi	0
*#n	0
*[high][labial]	0.08
*m[high]	0
*#[nasal]	3.87
*[labial][high]	0.05
*[high]b[high]	1.57
*i[labial]i	0
*u[labial]i	0
*i[labial]i	0
*[high][{p,m}][high]	2.26

32 independent runs of the simulation were performed, and it is the 32 grammars resulting from these runs that will be analyzed. For the purposes of analysis, I will be interested in only one aspect of these grammars: what type of units do the constraints in the grammar refer to – segmental units or feature labels? To isolate this aspect, I extracted, for every grammar, all constraints that had non-zero weight in that grammar, and I inspected their definitions. Only constraints with non-zero weight were included, because constraints with zero weight have no influence whatsoever on the outcome of the grammar, and the grammar would have had the same effect if these constraints did not

exist. Since grammars were continuously updated with new constraints, and new constraints could reset old constraints to zero (as explained in section 2.3.2.5), but constraints were never removed, some constraints in the final grammar had zero weight . The grammar displayed in Table 10 above yields the following constraints with non-zero weight:

Table 11. Non-zero constraints from Table 10

Constraint	Weight
*m#	2.45
*[high][labial]	0.08
*#[nasal]	3.87
*[labial][high]	0.05
*[high]b[high]	1.57
*[high][{p,m}][high]	2.26

This grammar (which is the grammar generated at the first run of the simulation) was represented as the following set of constraints for the purpose of analysis:

(30) The set of constraints in Table 11
 {*m#, *[high][labial], *#[nasal], *[labial][high], *[high]b[high], *[high][{p,m}][high]}

The non-zero-weight constraints for each grammar were subsequently sorted according to which of the three phonotactic patterns of the toy language (see section 2.2.3) they encoded:

(31) The three phonotactic patterns
 a. no final [m]
 b. no initial nasals
 c. no labials between high vowels

Most constraints clearly represented one of these three patterns. However, 8 out of 32 grammars also included one of the following two constraints¹⁷:

(32) [m] plus vowel constraints
 a. *Vm (no vowel followed by [m])
 b. *mV (no [m] followed by a vowel)

¹⁷ 4 grammars had *Vm; 4 grammars had *mV.

These constraints do not represent a unique pattern among the three listed in (31) above. Rather, (32a) partially represents patterns (31a) and (31c). Word-final [m] always occurs after a vowel (since only CVCVC word shapes were considered), but not every postvocalic [m] is word-final. [m] is one of the labial sounds that may not occur in between high vowels, and V_ is a partial description of that context.

In a similar way, *mV (= (32b)) is a partial description of patterns (31b) and (31c). Every word-initial [m] (which is one of the nasals, that are prohibited word-initially) is also prevocalic, but not every prevocalic [m] is word-initial. Every [m] in between two high vowels is prevocalic, but not every prevocalic [m] is in between two high vowels.

Since these 2 constraints do not fully represent any of the three patterns in the data set, they will be excluded from the following discussion, which will focus on the shape of constraints per phonotactic pattern. Specifically, I will look at how often features and unanalyzed segments were used to encode these patterns. I will first examine the word-final pattern (“no word-final [m]”), then the word-initial pattern (“no word-initial nasals”), and, finally, the word-medial pattern (“no labials in between two high vowels”).

The complete results of the 32 runs of the simulation can be found in Appendix B.

2.4.1 No word-final [m]

The pattern that prohibits word-final [m] was always represented in the grammars with one or both of the following two constraints:

(33) Constraints for “no word-final [m]”

- a. *m# (penalty of 1 for every word-final segment which is [m])
- b. *[labial,nasal]# (penalty of 1 for every word-final labial nasal segment)

However, there was a very strong preference for using *m#. 30 out of 32 grammars only had the segment-based constraint *m#, not the feature-based *[labial,nasal]#. One grammar had both *m# and *[labial,nasal]#, and it was just the one remaining grammar that had *[labial,nasal]# without *m#.

Thus, we can say that 30 grammars encoded this pattern with reference only to segments, 1 grammar encoded it with reference both to features and segments (where the feature-based constraint and the segment-based constraint were weighted equally), and 1 grammar encoded it with reference only to features. This reveals a strong bias to represent the one-segment pattern “no word-final [m]” with segment-based constraints.

Comparison between this pattern and the other two patterns will reveal if this tendency towards segmental representation is unique to the “no final [m]” pattern, as was predicted by the emergent feature model (section 2.2.4).

2.4.2 No word-initial nasals

The pattern that bans word-initial nasals [m, n, ŋ] in the toy language was represented by the constraint *#[nasal] in 29 out of 32 grammars. In 3 of these 29 grammars, this constraint was accompanied by *#[nasal]V.

(34) Most frequent representations for “no word-initial nasals”

- a. *#[nasal] (penalty of 1 for every word-initial nasal segment)
- b. *#[nasal]V (penalty of 1 for every word-initial prevocalic nasal segment)

These two constraints are homonymous for the forms examined here, since all words have the shape CVCVC.

The remaining 3 (out of 32) grammars use feature labels which do not correspond

to traditional features: [$\{m, \eta\}$] (a feature assigned to $[m]$, $[\eta]$, but not $[n]$) and [$\{n, \eta\}$] (a feature assigned to $[n]$, $[\eta]$, but not $[m]$)¹⁸. The constraints in these grammars are shown in Table 12 below.

Table 12. Grammars that appeal to strange “features”

Run #	Constraints representing word-initial pattern
11	*#m *#[$\{n, \eta\}$] *#[$\{n, \eta\}$]V
16	*#[$\{m, \eta\}$] *#[$\{n, \eta\}$]
17	*#m *#[$\{n, \eta\}$]

Of these three deviant grammars, only the grammars at runs 11 and 17 employ a segment-based constraint (*#m) – and even then it is in conjunction with at least one feature-based constraint.

From these data, we may conclude that there is a strong bias toward representing the pattern which prohibits word-initial nasals with features. None of the 32 grammars represented the pattern exclusively with segment-based constraints, and only 2 grammars represented the pattern with a combination of a segment-based constraint and one or more feature-based constraints (see runs 11 and 17). The other 30 grammars refer only to features with respect to the word-initial restriction.

2.4.3 No labials between high vowels

Finally, the word-medial pattern (which penalized labial consonants in between high vowels) found a much more variant grammatical representation, owing to its

¹⁸ These features do correspond to intersections of phonological features proposed in the literature – but not to single features. $[m, \eta]$ can be described as $[+nasal, -coronal]$ in Chomsky and Halle’s (1968) system, or as $[nasal, peripheral]$ according to the systems proposed by Dogil (1988) and Avery and Rice (1989). $[m, n]$ can be described as $[nasal, lingual]$ in Clements and Hume’s (1995) system.

complexity.

17 out of 32 grammars represented this pattern with the constraint *[high][labial][high], as was expected. In 3 of these 17 grammars, *[high][labial][high] co-occurred with the constraint *[high][labial], and in 1 of the 17, *[high][labial][high] co-occurred with *[labial][high].

(35) Most frequent representation for “no labials between high vowels”

- a. *[high][labial][high]
(penalty of 1 for every sequence of a high vowel, a labial consonant and another high vowel)
- b. *[high][labial]
(penalty of 1 for every sequence of a high vowel followed by a labial)
- c. *[labial][high]
(penalty of 1 for every sequence of a labial followed by a high vowel)

In any event, these 17 grammars all refer to features only. The remaining 15 grammars had constraints of various shapes and sizes: both two-position and three-position constraints ((35a-b)), and constraints which referred to segments only, features only, or combinations of features and segments ((35c-e)).

(36) Other constraints for “no labials between high vowels”

- a. two-position constraint: *[high]m
- b. three-position constraint: *[high]m[high]
- c. segments-only constraint: *mu
- d. features-only constraint: *[high][{p,b}][high]
- e. features-and-segments constraint: *u[labial][high]

Although there was considerable variation in the constraints representing the word-medial pattern in these 15 grammars, there was one clear generalization: none of these grammars represented the pattern only in terms of segment-based constraints. The table below provides some examples of how the word-medial pattern was represented in these grammars:

Table 13. Other constraints for “no labials between high vowels”

Run #	Constraints representing word-medial pattern
8	*[high]m; *[high][{p,b}][high] *[high][labial]i *[high][labial]u
14	*[high]m[high] *[high][{p,b}][high]
27	*mi *mu *[high]m *u[{p,b}][high] *[high][{p,m}][high]

Summarizing, none of the grammars represented the word-medial pattern in terms of segment-based constraints only; 15 out of 32 grammars represented the pattern while appealing to a mixture of segments and features, and the 17 remaining grammars represented the pattern with appeal to features only.

2.4.4 Summary of results

The three subsections above showed clear differences in grammatical representation between the three phonotactic patterns in the toy language. The table below summarizes these differences – with type of units appealed to (segments, segment/features, features only) tabulated against phonotactic pattern:

Table 14. Type of unit appealed to for each phonotactic pattern

Phonotactic restriction	Constraints only appeal to segments: # of grammars	Constraints appeal to mixture of segments and features: # of grammars	Constraints only appeal to features: # of grammars
no word-final [m]	30	1	1
no word-initial nasals	0	1	31
no labials between high Vs	0	15	17

Table 14 shows that there was an overwhelming preference for the word-final pattern, which appeals to the single segment [m], to be represented in terms of the segment [m] only. At the same time, neither of the two other patterns, both of which are based

exclusively on multi-segment classes, are represented with segments only in any of the grammars.

While the word-initial pattern has an almost absolute preference for being represented with features only, the word-medial pattern is represented with features only in about 1 in 2 runs of the simulation. This latter fact can be attributed to the length and complexity of the constraint *[high][labial][high], in which all three slots are occupied by features.

In any event, there is a clear asymmetry between the one-segment pattern (“no final [m]”) and the two multi-segment patterns (“no initial nasals”, “no labials in between high vowels”): the one-segment pattern is almost never represented with a feature in the constraint (only 2 out of 32 runs), whereas the multi-segment patterns are always represented with a feature in the constraint (32 out of 32 runs). This matches the intuition given in section 2.2: a truly emergentist model of feature learning leads to a grammar which appeals to a spectrum of levels of abstraction (such as segments/phones and features), instead of a grammar which always appeals to the highest available level of abstraction (classificatory phonological features).

2.5 Discussion and conclusion

This chapter introduced a radically emergentist model of phonological feature induction. This model has induction of phonological features alongside induction of OT-style constraints – a combination that had not been explored before. The model had as its goal to create grammars with maximally general constraints – following applications of the same principle in Albright and Hayes (2002, 2003) and Hayes and Wilson (2008). This created a model in which the presence of phonological features is motivated by the

learning of grammar itself: phonological features are one of the instruments by which constraints can be made more general. This is why features in this model are truly emergent.

One reason why this model is interesting is because it sees features as truly emergent – in the sense that their presence is motivated by external factors (namely, generality in the grammar's constraints). The canonical view, in which the vocabulary for segmental phonology (in the form of phonological features) is predefined and innate (see, for instance, Chomsky and Halle 1968), allows only for grammars which refer to features in constraints which represent segmental patterns. However, I showed through computational implementation of the radically emergentist model proposed here that this model predicts grammars which refer to a mixture of various levels of abstraction. In the case investigated here, these levels of abstraction were unanalyzed segment (allophone/phoneme) units and phonological features. Such grammars emerged from this learning model because this model's goal is to learn grammars with maximally general and concise constraints, independently of the level of representational abstraction (segment, feature, ...) to which these constraints refer.

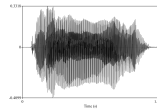
Grammars referring to different levels of abstraction in phonological representation have interesting properties in and of themselves. For instance, the segment and the feature in the current model can be seen as distinct levels of representation. One and the same sound event could be described as either [m], or as [labial, nasal], and a constraint can refer to either mode of description, as shown in Figure 6 below:

Figure 6. Different levels of abstraction at which a sound transcribed as [m] can be represented

featural representation: [labial, nasal]

segmental representation: [m]

acoustics:



The current model does not allow mismatches between segments and features (i.e., a segment [m] is always attached to [labial] and [nasal], and *vice versa*). However, segment units, in this model, can exist without features (as in the initial state of the learner, for instance). Moreover, the feature representations for each segment are learned in the process of learning the grammar. These ideas together suggest that, conceptually speaking, the mappings between features and segments should be represented by constraints in the grammar.

The current implementation only represented these mappings by storing for every feature which segments were associated with it. The constraints that refer to a feature were given violation marks based on these sets of segments. However, the assignment of features to segments was never regulated by constraints in the grammar.

A more extended model in which the mappings between segments and features are represented in the grammar would be forced to implement these mappings as violable constraints of the sort displayed in (37). A very similar kind of constraint has been proposed and formalized by Boersma (1998, 2007), Escudero and Boersma (2003, 2004), Boersma and Escudero (2008) in the form of Cue Constraints, which are violable statements that regulate the possible and impossible mismatches between phonological (featural) units and their phonetic expression in a language.

(37) Constraints for regulating segment/feature matching in a more extended model

a. $x \in \{[p], [b], [m]\} \rightarrow \text{labial}(x)$: One violation mark for every segment which is one of $\{p, b, m\}$ and does not have the feature $[\text{labial}]$.

b. $x \in \{[m], [n], [\eta]\} \rightarrow \text{nasal}(x)$: One violation mark for every segment which is one of $\{m, n, \eta\}$ and does not have the feature $[\text{nasal}]$.

In a more extended model that contains this type of constraints, the GEN component of the grammar will have to consider candidates in which the intended segment/feature pairings are not respected. Some examples are given in (38):

(38) Examples of mismatching representations in a more extended model

a. $[\text{nasal}]$ ([m] is not assigned the feature $[\text{labial}]$)

|
[m]

b. $[\text{labial}]$ ([b] is assigned $[\text{nasal}]$)

| $[\text{nasal}]$
| /
[b]

Even though the representations in (38) violate the intended segment/feature mappings, there are certain constraint rankings under which such representations might receive high probability. For instance, (37a) might receive higher probability than a candidate in which $[m]$ is classified as both $[\text{labial}]$ and $[\text{nasal}]$ in a grammar in which a constraint against the feature $[\text{nasal}]$ in a certain context far outranks constraint (37a).

The conceptual possibility of mismatches between independent levels of representation is reminiscent of such models as Turbidity Theory (Goldrick 2001), Colored Containment (Oostendorp 2008), Abstract Declarative Phonology (Bye 2006) and Bidirectional OT (Boersma 2007, 2011). Models of this kind allow for accounts of phonological opacity (Kiparsky 1971, 1973) without the use of extrinsic derivational ordering (see, for instance, Goldrick 2001, Bye 2006, and Boersma 2007 for examples of such accounts).

One final note is due with regard to the multi-leveledness of the representations that arise in these simulations. Since both segments and features are independent parts of the surface representations that emerge, Richness of the Base (Prince and Smolensky 1993) requires that the grammar pick a grammatical surface representation for underlying forms that have /m/ without a corresponding feature specification, as well as underlying representations that only have a feature specification [nasal, labial] but no segment /m/ to go with it. Such underlying representations are given in (39).

(39) “Mismatching” underlying representations

- [nasal]
|
[labial]
- a. /m a n i b/
(first segment is /m/ but does not have a [labial, nasal] specification)
- [nasal] [nasal]
[labial] | [labial]
- b. / a n i b/
(first segment has a feature specification but no segment specification)

Since the current grammar model has the mapping between [nasal, labial] and /m/ specified outside the grammar, every input in which [labial, nasal] occurs without /m/, or *vice versa*, would only be mapped onto outputs in which [m] and [labial, nasal] go hand in hand.

In the more extended model sketched in the paragraphs above (where the mapping between segments and features is regulated by violable constraints), inputs such as the ones in (39) could potentially surface as fully faithful outputs, if the constraints regulating the link between [labial, nasal] and [m] are sufficiently low-ranked. However, such a low ranking would only be justified by some kind of opaque phenomenon in which outputs with [m] but no [labial, nasal] specification (or [labial, nasal] but no [m] specifications)

are somehow needed. Otherwise, the constraint that regulates the co-occurrence of [labial, nasal] and [m] would be high-ranked and would force every winning output for the inputs in (39) to have [labial, nasal] wherever [m] occurs, and *vice versa*.

Another interesting property of grammars like the ones induced in this model is that they call to mind the existence of evidence from work in speech production and perception (e.g., McQueen, Cutler, and Norris 2006, Jesse 2007, Nielsen 2011) that speech is processed at multiple distinct levels of abstraction (exemplars, segments, features). The techniques used in this work (which all have to do with asking subjects to expand a pattern presented to them in training data) could also be applied to test the level of abstraction referred to by various constraints in the grammar.

The latter means that the language-internal predictions of the current emergentist model are testable through behavioral experiments. This is an innovation for models with emergent explicit grammatical structure – which can be contrasted with exemplar models, in which grammatical structure is emergent but implicit (see, for instance, Wedel 2003, 2011). Models such as Mielke’s (2004) have mainly relied on typological patterns, but this model allows for potential experimental evidence in favor of the emergent feature scenario.

One example of such a behavioral test of the within-speaker predictions of the model would be to combine the methodology of the studies mentioned above with Halle’s (1978) “Bach-testing”, briefly discussed in section 2.2.4, where it was pointed out that a slight extension of the current model that would assign novel segments to features in the model as a function of phonetic similarity to the segments that are already assigned to each feature makes such a “Bach test” possible. Lacking such an extension, one might

simply extend each feature to all novel segments that have the phonetic properties shared by the old segments that belong to that feature. For instance, if feature [F] is assigned to [p,b,m], then, in a pencil and paper analysis, all other segments that have a labial gesture could be assigned [F] as well – even though this might be only a rough approximation.

Given some way of assigning features to novel segments, a potential example of a “Bach test” for the hypothesis that single segment phonotactic generalizations are formulated with segments rather than features would involve confronting speakers of English with a novel segment which conforms to the feature specification of [s], which is the only segment allowed as the first consonant in #CCC clusters – see section 2.2.2.

Since one might expect that the pattern should be formulated in its most concise form, the features that match [s] in the constraints are expected to be [-voice,+anterior,+strident]. An example of a novel (non-English) segment that matches that description is [s̺] (dental [s]). The question is whether speakers that are taught this new segment unit [s̺] will automatically extend the pattern to this segment (i.e., forms such as [s̺plit] will be as acceptable as [split]).

It is not important that [s̺] may have features that do not match those of [s] – for instance, [±distributed]. All that matters is that [s] and [s̺] have the same values for [±voice], [±anterior], and [±strident] – since these are the three features which are sufficient to distinguish /s/ from the other phonemes in English. The reason why only this matters is that I assume some economy mechanism which has the same effect as Chomsky and Halle's (1968) Evaluation Metric: every constraint only refers to features that are necessary to distinguish the sounds that the constraint applies to from those that it does not apply to. In this manner, the constraint *#^[-voice,+anterior,+strident]CC does

not care about the value of [\pm distributed] or [\pm dorsal] of its first segmental position.

The canonical innate feature model predicts that the grammatical statement of the pattern of only-[s]-starting-#CCC must appeal to features only, and thus, speakers will automatically extend the pattern to the novel segment (e.g., [ʂ]). On the other hand, the emergent feature model predicts that the single-segment domain (“only [s]”) makes it very likely that the pattern will be represented with [s] as a segment only, without featural specification. This means that there will not be a strong impetus to generalize the pattern, so that the novel segment ([ʂ]) will not be allowed at the beginning of #CCC clusters.¹⁹

This and similar experimental methodologies, when applied to this problem, will shed light on the predictions of both models. Consequently, behavioral tests of this sort are an important direction for future work.

In connection to such tests, it is also very important to develop a way to interpret the features learned by the algorithm phonetically, so that these labels can be extended to novel segments. Such algorithms can be created by including a corpus with some simplified encoding of acoustic (and/or articulatory) realizations of the segments that are classified by each feature label, and applying some form of linear regression to these data to determine which phonetic characteristics are the best predictors for the class of segments determined by each feature label.

¹⁹ One potential empirical objection to this prediction made by the emergent feature model is the fact that loanwords from Yiddish and German have been borrowed into English with [ʃ] as the first member of word-initial clusters that normally only allow [s] as their first member, such as [ʃmʌk], [ʃmuz], and [ʃpɪts]. However, this fact cannot decide between these two models. [ʃ] contrasts with [s] in English, and this contrast is not ignored in onset phonotactics: compare [sn-], [ʃɪ-] vs. *[ʃn-], *[ʃɪ-]. This means that the constraints on clusters like [sm] and [sCC] must be formulated to exclude [ʃ]: *[^][+cor,+ant,-voice,+str]CC. If English phonotactics already explicitly excluded [ʃ] in that position, then both the innate feature model and the emergent feature model fail to predict that allowing [s] in these clusters entails that [ʃ] should also be allowed.

Incorporating phonetic factors into the clustering process that finds features (2.2.3-2.2.4) would also be an important next step. Since the overwhelming majority of phonological classes in languages is phonetically consistent (see, for instance, Mielke 2004, 2007), a bias that pressures languages to generalize across phonetically similar segments is plausible. Incorporating phonetics into clustering would constitute such a bias.

Another direction for future research is to incorporate the learning of segment categories from acoustic input into the model pursued in this chapter. This acoustic input could also be used to find phonetic correlates for each feature found by the learner (see above).

Finally, it is very important to test this model on other, more extended cases of phonotactics. This will allow for validation of the result obtained here across different cases, and it might also reveal other properties that arise from an emergent feature model that have not been considered before.

This and other projects are of high importance for the results presented here. However, the current results in themselves provide a novel perspective on the problem of phonological abstraction and the dialogue between innate and emergent approaches to phonological structure.

CHAPTER 3

LEARNING OPACITY IN STRATAL MAXENT: OPACITY AND EVIDENCE FOR STRATAL AFFILIATION

This chapter is a revised version of “Learning opacity in Stratal Maximum Entropy Grammar”, co-authored with Joe Pater, currently under review for journal publication. My contribution to this paper includes co-designing and running simulations, interpreting the results, and writing sections 3.2-3.5.

3.1 Introduction

(Kiparsky 1971, 1973) draws attention to empirical evidence from historical change that suggest that at least some opaque phonological process interactions are difficult to learn. In his work on Stratal Optimality Theory (Stratal OT), he further claims that independent evidence about the stratal affiliation of a process is helpful in learning an opaque interaction (Kiparsky 2000). In this chapter, we develop a computationally implemented learner for a weighted constraint version of Stratal OT, and show that for an initial set of test cases, opaque patterns are indeed generally more difficult than their transparent counterparts, and that information about stratal affiliation does make learning of an opaque pattern easier.

In the Stratal OT approach to opacity (see also especially Bermúdez-Otero 1999, Bermúdez-Otero 2003), grammars are chained together to produce outputs that would not be possible of a single level. For example, the classic case of the opaque interaction between diphthong raising and flapping in Canadian English *mitre* (1) can be produced by raising in the first (word level) grammar, and then flapping in the second (phrase level) one. As we will see shortly, the final candidate cannot be produced by a single OT

grammar, with the standard set of constraints assumed for this case (though see Pater 2014).

(40) /maɪtə/ – Grammar 1 → [mʌɪtə] – Grammar 2 → [mʌɪrə]

This chaining of grammars results in a hidden structure problem (Tesar and Smolensky 2000), insofar as only the output of the final grammar, and not any earlier ones, is available in the learning data. Bermúdez-Otero (2003) develops a revision to the constraint demotion algorithm that is specifically tailored to the problem of hidden structure in Stratal OT. In this chapter, we instead apply a general approach to learning hidden structure in probabilistic OT to the specific case of chained grammars.

The variant of probabilistic OT adopted here is Maximum Entropy Grammar (Goldwater and Johnson 2003, Hayes and Wilson 2008), which defines the probability distribution over a candidate set in a conveniently direct way: candidate probabilities are proportional to the exponential of the weighted sum of violations. This is illustrated in the tableau in (41), in which weights are indicated under constraint names (which are simplified versions of the ones used for the case study in section 3.3.2). The weighted sum appears in the column headed “ H ”, for Harmony (Smolensky and Legendre 2006). The exponentiation of H appears under e^H , and the resulting probability under p .

(41) Computation of probabilities over output candidates in MaxEnt

/maɪtə/	*art 7	Ident(C) 5	Ident(V) 1	*VTV 0	H	e^H	p
a. maɪtə	–1			–1	–7	0.001	0.002
b. maɪrə		–1			–5	0.007	0.018
c. mʌɪtə			–1	–1	–1	0.368	0.973
d. mʌɪrə		–1	–1		–6	0.002	0.007

The *art constraint penalizes the unraised diphthong [aɪ] before a voiceless consonant, and Ident(V) penalizes raising to [ʌɪ]. With *art having greater weight than Ident(V),

raising before a voiceless stop, as in (41c), has greater probability than the faithful (41a.). *VTV penalizes an alveolar stop between vowels, and Ident(C) penalizes the change from stop to flap. With higher weighted Ident(C) than *VTV, the probability on flapping candidates (41b) and (41d) is lowered. Candidate (41d) is the correct output in Canadian English, but it has a proper superset of the violations of (41b), since flapping also satisfies *ait.

We assume that constraints always have non-negative weights, so that their typological predictions are not reversed (see also Prince 2003, Pater 2009, Potts, Pater, Bhatt, Jesney, and Becker 2010, Boersma and Pater 2016). There is no non-negative weighting of the constraints that will give [mʌɪrə] the highest probability in the tableau. In this respect, the situation is the same as in single-level OT, in which (41d) cannot be optimal (since it is harmonically bounded by (41b)). In Figure 9 in section 3.3.2, we will see that [mʌɪrə] can emerge from a second chained MaxEnt grammar with highest probability.

One peculiarity of MaxEnt is that (41d) can tie with (41b): when Ident(C) and Ident(V) both have a weight of 0, (41b) and (41d) are assigned equal probability (see Pater 2016 for further discussion and references on harmonic bounding and probabilistic weighted constraint theories). Ties like these make it so that certain unmotivated processes can occur optionally. For instance, in the local minimum shown in Table 17 in section 3.2.1, the mapping /e/ → [ɛ] optionally occurs in an open syllable, even though no constraint in the analysis prefers this mapping in that context, but instead all constraints prefer /e/ → [e] in that context. Similarly, in the first two local minima shown in Table 24 in section 3.3.3, the mapping /aɪ/ → [ʌɪ] optionally occurs before a voiced consonant [d],

even though none of the constraints prefer this mapping in that context. In both cases, this occurs because of ties between candidates that arise from zero weight on the constraints that regulate vowel quality.

Our way of dealing with the hidden structure problem posed by Stratal OT is an extension of the approach to Harmonic Serialism in Staubs and Pater (2016). The probability that the grammar assigns to an output form is the sum of the probabilities of the derivational paths that lead to it from its input. The probability of a derivational path is the product of the probabilities of each of its component steps, since by the definition of Stratal OT, computation at a any derivational stage may not look back further than its own input (see also section 3.2.1); the probabilities of all derivational paths that start from the same input sum to 1. The probability of each step in a derivational path is determined by the weights of the constraints that apply at that level (word or phrase; see section 3.2.1 for more details). We use a batch learning approach in which the objective is to minimize the divergence between the predictions of the grammar and the learning data, subject to a regularization term that lower weights. This is a standard approach to learning MaxEnt grammar (see e.g. Goldwater and Johnson 2003, Hayes and Wilson 2008).

Jarosz (2006a, submitted) shows how considering every possible hidden structure (with a probability attached to every such structure) in generating an overt candidate can be applied to phonology in her work on Expectation Maximization with Optimality Theory. Here, we follow Eisenstat (2009), Pater et al. (2012), Johnson et al. (2015), and Staubs and Pater (2016) in adopting a weighted constraint variant of this general approach.

When there is hidden structure, the learner is not guaranteed to converge on the global optimum, that is, a set of weights that best satisfy the objective function (the solution space is not guaranteed to be convex; see, for instance, Quattoni et al. 2007). Instead it may only find a local optimum, a set of weights that is the best within the space that the learner can explore. We provide examples of such local optima in Table 17 and Table 18 in section 2.3, and Table 24 and Table 25 in section 3.3.

A local optimum is defined as a constraint weighting whose value on the objective function (see (49) for a statement of the objective function used here) is not the overall lowest that is possible within the space of constraint weightings, but the (estimated) first-order derivative of the objective function still equals 0 (i.e., moving anywhere locally in the space of possible weightings will result in a higher value on the objective function). The global optimum is the constraint weighting that does have the overall lowest objective function value. For the problem explored here, the global optimum is reached whenever the learner matches the pattern in the training data, since the objective function chosen here is minimized whenever the predictions of the grammar match the probability distribution in the training data, and the learner is based on a procedure that finds optima (see section 3.2.2 for more details). Therefore, if there is a constraint weighting that generates the pattern in the training data, all outcomes of the learner that do not match the pattern in the training data are automatically local optima.

A simple and standard way of finding a global optimum in such cases is to try different starting positions for the learner (initializations), with the hope that one or more will allow the learner to find the global optimum. In the limit, as long as we are sampling initializations from the space in which optimal solutions exist, this will allow the learner

to find a global optimum. In our test cases, there were always runs of the learner that granted more than 0.5 probability to the overt forms that had highest probability in the learning data, our criterion of success. Our measure of “ease of learning” for the various patterns we study is the probability that a learner will be successful by this criterion, over a set of random initializations.

In the next section we provide more details about our grammatical and learning theories, in the context of our first test case, a relatively simple example of opaque tense-lax vowel alternations in French. In section 3.3 we turn to the opaque interaction of diphthong raising and flapping in Canadian English, and the effect of supplying evidence to the learner of phrasal non-raising. Section 3.4 provides an exploration of the mechanisms that lead to incorrect learning of (predominantly) opaque patterns. Finally, overall conclusions and directions for further research can be found in section 3.5.

3.2 Case study I: Southern French tensing/laxing

3.2.1 Data and analysis

As reported by Moreux (1985, 2006), Rizzolo (2002), and Eychenne (2014), certain dialects of Southern French have a generalization that mid vowels are tense ([e, o, ø]) in open syllables, while they are lax ([ɛ, ɔ, œ]) in closed ones, as shown in (42a).²⁰ This generalization, conventionally called *loi de position* (“law of position”), is enforced by change in the laxness of the vowel, as illustrated in (42bc). This process applies within words.

²⁰ Mid vowels are also lax in open syllables followed by a schwa syllable; this complication will be disregarded here, and the reader is referred to Selkirk 1978, Moreux 1985, and Rizzolo 2002 for potential explanations of this.

(42) Examples of *loi de position* (Eychenne 2014, Rizzolo 2002)

- a. /sɛl/ → [sɛl] ‘salt’
cf. /sɛ/ → [sɛ] ‘knows’
- b. /pøʁ/ → [pøʁ] ‘fear’
cf. /pøʁ-ø/ → [pø.ʁø] ‘easily frightened’
- c. /pøʁ/ → [pøʁ] ‘pore’
cf. /pøʁ-ø/ → [po.ʁø] ‘porous’

Tensing/laxing is made opaque (is counterbled) by resyllabification across word boundaries, at least in some cases.²¹ (43b) below shows that, even though the second syllable of the phrase [kã.pø.ʁã.ʁa.ʒɛ] is open, its vowel is a lax [œ] rather than changing it to tense [ø] (as it happens in the pair in (42b)).

The symbol “#” will be used throughout this paper to stand for a word boundary that is impermeable to phonology. At the word level, phonology cannot look across words to see which segments are adjacent, which, in (43b) is indicated by an impermeable word boundary in the intermediate (word level) output: kã.pøʁ # ã.ʁa.ʒɛ.

(43) Opaque interaction between resyllabification and tensing/laxing (Rizzolo 2002:51)

- a. /kãpøʁ/ → [kã.pøʁ] ‘camper’
- b. /kãpøʁ ãʁaʒɛ/ → kã.pøʁ # ã.ʁa.ʒɛ → [kã.pø.ʁã.ʁa.ʒɛ] ‘enraged camper’

In this simulation, we investigated whether our learner could deal with this simple case of opacity. This also serves as a simple case to illustrate in more detail the functioning of the learner. The data that we offered to the learner consisted of exceptionless *loi de position* as well as exceptionless opacity through resyllabification:

(44) Opaque interaction, as in actual Southern French

- a. /sɛt a/ → (sɛt. # a →) [sɛ.ta] ‘this (letter) A’ <cette A>
- b. /sɛ ta/ → (sɛ. # ta →) [sɛ.ta] ‘it is “ta”’ <c’est “ta”>

The constraint set that used for this subset of real French was maximally small, as shown

²¹ Rizzolo (2002:51) reports that this effect is not unexceptional, at least in the dialect he describes. In our case study, we will investigate an idealized version of this pattern in which the opaque interaction is unexceptional.

in (45): one constraint that prefers tense vowels in open syllables, one that prefers lax vowels in closed syllables, and a Faithfulness constraint. In order to be able to restrict the learning problem to just these three constraints, it was assumed that the constraints NoCoda and Onset were undominated (or had very high weight) at both derivational levels. This entails that consonants will be syllabified with the following vowel whenever such a vowel is available (*[set.a]), but whenever there is a consonant at the end of a word in a word level representation, this consonant will still be syllabified with the preceding vowel, since there is no following vowel that is visible (*se.t # a). In future work, NoCoda and Onset should also be included in the set of constraints whose weighting must be learned.

(45) Constraint set used for the Southern French case study

- a. *[-tense]/Open : One violation mark for every [-tense] segment in an open syllable.
- b. *[+tense]/Closed : One violation mark for every [+tense] segment in a closed syllable.
- c. Ident(V) : One violation mark for every vowel which is not identical to its input correspondent.

The original formulation of Stratal OT (Kiparsky 2000, Bermudéz-Otero 1999) allows for three levels of derivation: a stem level, a word level, and a phrase level – which all have the same constraints, but different rankings or weightings. For this problem, we will only consider two levels – a word level and a phrase level grammar. The word level grammar evaluates each word individually, without regard to its neighbors in the phrase. By contrast, the phrase level, which operates on the output of the word level grammar, does evaluate entire phrases together.

With this setup, then, the opaque interaction is obtained when word level *[+tense]/Closed and *[-tense]/Open have high weight and word level Ident(V) has low weight, while the opposite holds at the phrase level. This corresponds to activity of the *loi*

de position at the word level, and its inactivity at the phrase level.

This latter scenario is the only option to derive this opaque interaction in Stratal OT. As will be shown in tableau (47) below, Phrase level Markedness disprefers surface [sɛ.ta], so that the mapping from /e/ to [ɛ] cannot be derived at the phrase level. At the same time, word level Markedness prefers the corresponding word level output, set # a, because the first vowel is in a closed syllable at that level, because the phrasal context is invisible. This can be seen in tableau (46a) below.

Activity of *loi de position* at the word level can be expressed with the following weights (found in successful runs of the learner described in section 3.2.2):

(46) a. Word-level derivation for “this A”

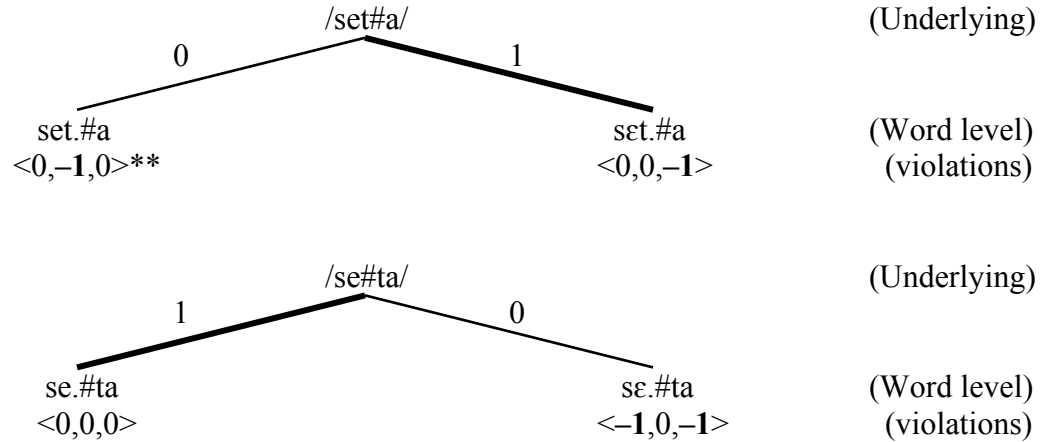
/set#a/	*[-tense]/Open 6.24	*[+tense]/Cl 6.24	Ident(V) 0	H	e ^H	p
set # a		−1		−6.24	0.002	0.00
set # a			−1	0	1	1.00

b. Word-level derivation for “it is ‘ta’”

/se#ta/	*[-tense]/Open 6.24	*[+tense]/Cl 6.24	Ident(V) 0	H	e ^H	p
se # ta				0	1	1.00
se # ta	−1		−1	−6.24	0.002	0.00

Figure 7 shows one-level derivations in (46) in the form of a derivation graph. In these graphs, the various possible derivational paths for an input are given in the form of a tree. Each branch is a derivational step, and thus has a probability associated with it, which is given next to the branch. Branches whose probability is close to 1 are shown thicker, to indicate that derivations will almost always proceed along these paths. The violation vectors for each candidate in the derivation are given under that candidate in the form of an ordered set < >. The order in which each constraint occurs in these vectors is given under the graph.

Figure 7. Word level derivation graphs for /set#a/ and /se#ta/



**Violation vectors given in the order <*[−tense]/Open, *[+tense]/Closed, Ident(V)>

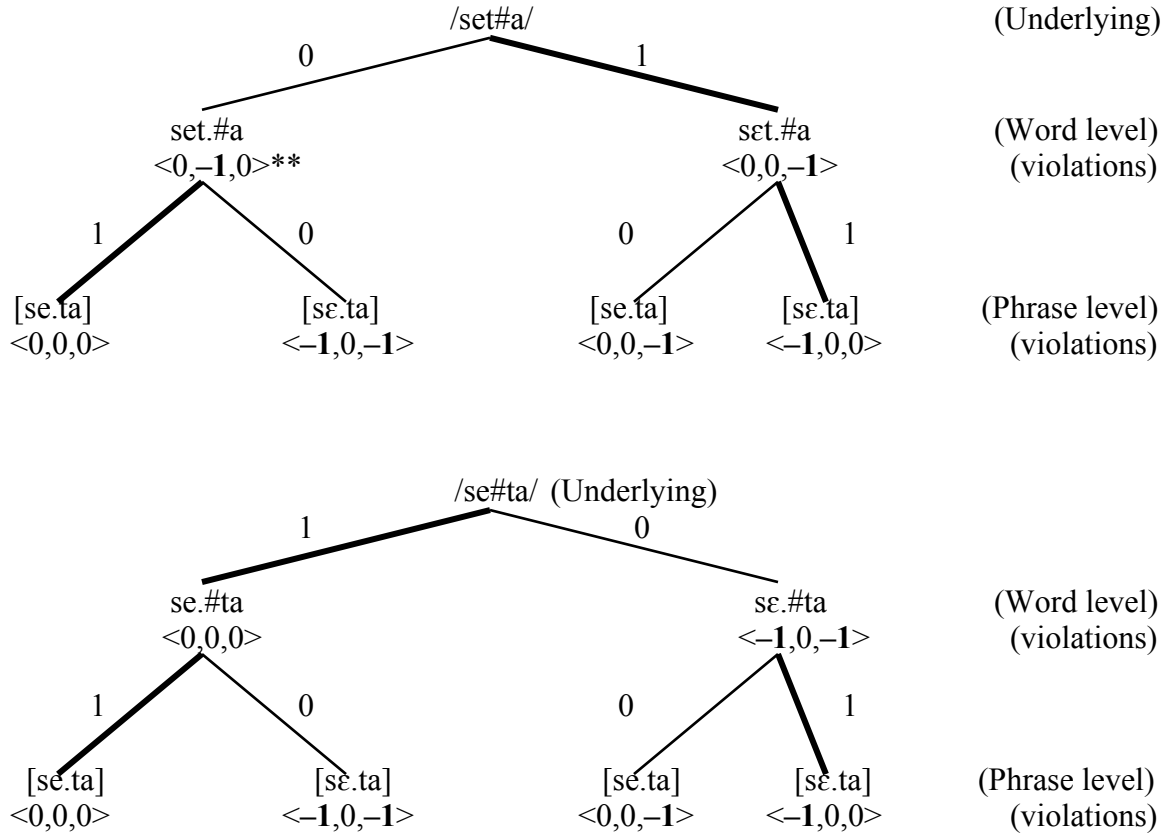
At the phrase level, giving high weight to Ident(V) and zero weight to both Markedness constraints results in 1.00 probability for all faithful mappings. This is illustrated in (47) for the phrase level derivation that takes place if the word level output is set. # a: phrase level outputs retain the [ε] that was created by closed syllable tensing at the word level with a probability of 1.00.

(47) Phrase-level derivation for “this A”

set. # a	*[−tense]/Open 0	*[+tense]/Cl 0	Ident(V) 6.93	H	e ^H	p
se.ta			−1	−6.93	0.001	0.00
se.ta	−1			0	1	1.00

Returning to the graph in Figure 7 above, the fact that faithful mappings win at the phrase level can be represented by the derivation graphs in Figure 8 below. As before, constraint violations are notated as a vector under each candidate, and the order in which violations are displayed is given under the graph itself.

Figure 8. Graphs for the complete derivation of “this A” and “it is ‘ta’”



**Violation vectors given in the order <*[-tense]/Open, *[+tense]/Closed, Ident(V)>

As can be seen in Figure 8, the single-stratum mappings shown in (46), (47), and Figure 7 are assembled into derivational paths which lead from an underlying representation (UR) to a surface representation (SR). The probability of a derivational path is obtained by multiplying the probabilities of every step in the path, as illustrated in (13). This is because every derivational step in Stratal OT is, by definition, independent from earlier or later derivational steps (Bermúdez-Otero 1999, Kiparsky 2000).

(48) Probabilities of derivational paths

a. $p(/set\#a/ \rightarrow set\#a \rightarrow [se.ta]) =$

$$p(/set\#a/ \rightarrow set\#a) \times p(set\#a \rightarrow [se.ta]) = 0.00 \times 1.00 = 0.00$$

b. $p(/set\#a/ \rightarrow set\#a \rightarrow [se.ta]) =$

$$p(/set\#a/ \rightarrow set\#a) \times p(set\#a \rightarrow [se.ta]) = 1.00 \times 1.00 = 1.00$$

The expected probability of a UR/SR pairing, then, is the sum of the probabilities of all derivational paths that lead to the surface form, as in Table 15. This table shows that, given the weights shown in (46) and (47), the desired output candidates (cf. (44)) are generated with a probability of 1.00.

Table 15. Expected probabilities of UR/SR pairings: sum over all derivational paths

Underlying	Derivational path	Probability	Surface	Probability
/set#a/	/set#a/ → set # a → [se.ta]	0.00	[se.ta]	0.00
	/set#a/ → set # a → [se.ta]	0.00		
	/set#a/ → set # a → [sɛ.ta]	0.00	[sɛ.ta]	1.00
	/set#a/ → set # a → [sɛ.ta]	1.00		
/se#ta/	/se#ta/ → se # ta → [se.ta]	1.00	[se.ta]	1.00
	/se#ta/ → se # ta → [se.ta]	0.00		
	/se#ta/ → se # ta → [sɛ.ta]	0.00	[sɛ.ta]	0.00
	/se#ta/ → se # ta → [sɛ.ta]	0.00		

Now that we have explained and illustrated the generation of probabilities over UR/SR pairings given the successful weights in (46,47), we will explain the structure of the learner that arrived at these weights in section 3.2.2. Then, in section 3.2.3, we will show how often this learner arrives at this successful set of weights within a sample of 100 runs, and what happens within that same sample when the learner does not arrive at this analysis.

3.2.2 Learning

Staubs (2014b) provides an implementation in *R* (R Core Team 2013) of a variant of Staubs and Pater’s (2016) approach to learning serial grammars. We modified it minimally to allow different violations of the same constraint at different derivational stages, which is sometimes necessary in a Stratal framework.

For instance, the constraint *[+tense]/Closed in our Southern French simulation

will be violated at the word level by a sequence [set # a], because even though [t] and [a] could form a separate syllable, the word boundary between them prevents this. However, the same constraint remains unviolated at the phrase level for the same sequence, since the grammar can now look beyond word boundaries and evaluate entire phrases, so that [t] and [a] do form their own syllable: [se.ta].

Expected probabilities of UR/SR pairings – $p_{exp}(UR \rightarrow SR_i)$ – are computed given sets of candidates, violation vectors, and constraint weights, as described and illustrated in section 3.2.1. The computation of these probabilities includes marginalization over hidden structure, as described there.

In addition, observed probabilities of UR/SR pairings – $p_{obs}(UR \rightarrow SR_i)$ – are provided to the learner. Since we are working with categorical data, all UR/SR pairings found in the data were given a probability of 1, and all others were given a probability of 0. The learner then minimizes the Kullback-Leibler-divergence (KL-divergence; Kullback and Leibler 1951) of the expected probabilities from the observed probabilities, which is a measure of how closely the grammar has been able to fit the data.

(49) KL-divergence

For all UR/SR pairings $UR \rightarrow SR_{i=1}, \dots, UR \rightarrow SR_{i=k}$:

$$D_{KL}(P_{obs} \parallel P_{exp}) = \sum_{i=1}^k p_{obs}(UR \rightarrow SR_i) \times \log \left(\frac{p_{obs}(UR \rightarrow SR_i)}{p_{exp}(UR \rightarrow SR_i)} \right)$$

Since the observed probability of any UR/SR pairing is either 1 or 0, KL-divergence in this case simply becomes the negative sum of the natural logarithm of every attested UR/SR pairing's expected probability. Thus, minimizing KL-divergence for categorical data is equivalent to maximizing the attested data points' summed likelihood (see Jarosz 2006a, 2006b, Heinz and Koirala 2010 for more on Maximum Likelihood learning of

phonology).

(50) KL-divergence for categorical data equal to negative total log likelihood of attested data

For all **attested** UR/SR pairings $UR \rightarrow SR_{i=1}, \dots, UR \rightarrow SR_{i=m}$

$$D_{KL}(P_{obs} \parallel P_{exp}) = - \sum_{i=1}^m \log p_{exp}(UR \rightarrow SR_i)$$

We chose to formulate our learner in terms of KL-divergence, so that our learner could also easily be applied to data that exhibit variation of type that a certain UR allows several SRs, but one is more common than others – see, for instance, (Guy 2011) Coetzee and Pater (2011) for attested examples of this. As an abstract example, let us consider a situation with 4 candidates, $\{ABCD\}$, of which A and C are unattested, B occurs 20% of the time, and D occurs 80% of the time. In this case, B and D are the attested candidates, and negative log likelihood is minimized when B as well as D receive 0.5 probability – which does not match the attested distribution (0.2 on B, 0.8 on D). However, KL-divergence is minimized whenever B receives 0.2 probability and D receives 0.8 probability – which does match the observed distribution.

Minimization of this objective function is done by the L-BFGS-B method of optimization (Byrd et al. 1995). This is a pseudo-Newtonian batch optimization method which uses approximations of the first-order and second-order derivatives of the objective function to take incremental steps towards an optimum. The ‘optim’ implementation of this method in *R* was used, specifying a minimum of zero on all constraint weights.

To place an upper bound on the objective function, an L2 regularization term

(Hoerl and Kennard 1970) with $\mu = 0$ and $\sigma^2 = 9,000$ is added to it.²² This regularization term penalizes weights as they increase, keeping the optimal solution finite. This prior also drives weights down to zero for any constraint that does not increase fit to the training data when its weight is increased.

Summarizing, our algorithm calculates probabilities of derivations in Stratal MaxEnt, sums over them to get the expected probabilities of overt forms, and fits the observed distribution of UR-SR pairs by minimizing KL-divergence. The output of learning is a grammar that consists in a constraint weighting that defines probabilities over UR→SR mappings. These probabilities need an extra interpretation step for the purposes of evaluating success on categorical data: a language is considered to be learned if all UR→SR mappings attested in that language are assigned more than 0.5 probability by the grammar that the learner found.

3.2.3 Results

The learner described in section 3.2.2 was run 100 times on the French *loi de position* data, with every run starting from random starting weights for each constraint, drawn i.i.d. from a uniform distribution over $[0, 10]$. The results are as follows:

Table 16. Results for the Southern French data set

Data set	Learned successfully out of 100 runs
/set#a/ → [se.ta]	98
/se#ta/ → [se.ta]	

Thus, opaque *loi de position* was learned successfully for an overwhelming majority of start weights. However, there is still a slight chance of incorrect learning, which we

²² Variance (σ^2) is set to 9,000, because higher values (10,000) lead to constraint weights that tend towards infinity for the Canadian English dataset, discussed in section 3.3.

defined as a probability of 0.5 or less on at least one surface form that has a probability of 1 in the learning data (see also section 3.1). The 2 unsuccessful runs yielded grammars that had free variation between tense [e] and lax [ɛ], as in Table 17.

Table 17. UR-to-SR probabilities for the local optimum

/set#a/	probability	/se#ta/	probability
[se.ta]	0.50	[se.ta]	0.50
[sɛ.ta]	0.50	[sɛ.ta]	0.50

This is a local optimum that arises when all constraint weights are set to zero, as in Table 18. It is an optimum, because set of weights yields the smallest KL-divergence of the model's predictions from the actual data within the space the learner explores.

Table 18. Weights that generate the local optimum

Word level			Phrase level		
*[-tense]/Open	*[+tense]/Cl	Ident(V)	*[-tense]/Open	*[+tense]/Cl	Ident(V)
0	0	0	0	0	0

As will be explained in section 3.4.1, there is a tendency for phrase level Faithfulness to lower its weight whenever the weight initialization does not already predict the word level forms necessary to let the desired SRs win at the phrase level. At the same time, when phrase level Ident(V) reaches zero weight, any effect of closed syllable laxing from the Word level cannot be transmitted to the final output. There is no locally better solution than 0.50 probability for all candidates, and the all-zero weight version of that solution minimizes the penalty on the regularization term, which prefers smaller weights.

3.3 Case study II: Raising and flapping in Canadian English

3.3.1 Data

Our second case study is a classic case of opacity, attested in Canadian English (Joos 1942, Chomsky 1964:73-74, Chomsky and Halle 1968:342, Idsardi 2000 and references therein, Pater 2014): the interaction between Canadian Raising and flapping.

Low-nucleus diphthongs [aɪ, aʊ] raise to [Λɪ, Λʊ] before voiceless consonants, (51a), and coronal oral stops /t, d/ become a voiced flap [ɾ] in a variety of contexts (De Jong 2011) – among others, between two vowels if the first is stressed and the second is not, as in (51b).

The two processes interact in a counterbleeding way: the fact that flapping cancels the voicelessness of underlying /t/ does not prevent that /t/ from triggering raising. This is illustrated in (51c).

(51) Opacity in Canadian English

- a. /laɪf/ → [lΛɪf] ‘life’ cf. /laɪ/ → [lΛɪ] ‘lie’
 b. /kʌt-ə/ → [ˈkʌɾə] ‘cutter’
 c. /maɪtə/ → [mΛɪɾə] ‘mitre’ cf. /saɪdə/ → [saɪɾə] ‘cider’

In addition, raising is restricted to the word domain, as illustrated in (52a), while there is no evidence of such a restriction for flapping, (52b). This fits into a Stratal picture in which raising applies at the word level only, while flapping applies at the phrase level only – as illustrated in (53).

(52) Raising is word-bounded

- a. /laɪ#fɔɪ/ → [lΛɪ fɔɪ] ‘lie for’ *[lΛɪ fɔɪ], cf. [lΛɪf]
 b. /laɪ#tu/ → [lΛɪ ɾə] ‘lie to’ *[lΛɪ ɾə], cf. [mΛɪɾə]

(53) Sketch of Stratal analysis of flapping and raising

/maɪtə/	→	mΛɪtə	→	mΛɪɾə
Underlying		Word Level:		Phrase Level:
Representation		Raising only		Flapping only

The transparent counterpart to this pattern, claimed to be spoken by some Canadian English speakers (Joos 1942), is a language which also has both raising and flapping as in (51ab) and (52), but the application of flapping blocks the application of raising, as in (54), since the flap [ɾ] is not voiceless. However, the existence of this transparent dialect has been disputed in later literature (Kaye 1990).

(54) Transparent interaction between raising and flapping

/maɪtə/ → [maɪrə] ‘mitre’

/saɪdə/ → [saɪrə] ‘cider’

/laɪf/ → [laɪf] ‘life’

In the next subsection, we will investigate the learnability of the opaque and the transparent versions of the Canadian English data. We will consider the contrast between opacity and transparency, and between various possible datasets.

3.3.2 Simulation setup

Canadian English provides at least two pieces of independent evidence regarding the stratal affiliation of the opaque raising process: evidence for its application outside the flapping context, as in (51a), and evidence for its word-boundedness, as in (52a). Based on Kiparsky (2000), we predict the availability of such evidence to make the opaque interaction more learnable. To investigate this, we considered four sets of inputs, shown in the columns of Table 19, that represent gradations of independent evidence regarding the opaque process. Both the transparent and the opaque interaction between raising and flapping was considered for these sets of inputs, yielding 8 data sets in total.

Table 19. Datasets for Canadian English

	‘mitre-cider’	‘mitre-cider-life’	‘mitre-cider-lie-for’	‘mitre-cider-life-lie-for’
Opaque	/maɪtə/ → [maɪrə] /saɪdə/ → [saɪrə]	/maɪtə/ → [maɪrə] /saɪdə/ → [saɪrə] /laɪf/ → [laɪf]	/maɪtə/ → [maɪrə] /saɪdə/ → [saɪrə] /laɪ#fɔɪ/ → [laɪ fɔɪ]	/maɪtə/ → [maɪrə] /saɪdə/ → [saɪrə] /laɪf/ → [laɪf] /laɪ#fɔɪ/ → [laɪ fɔɪ]
Transparent	/maɪtə/ → [maɪrə] /saɪdə/ → [saɪrə]	/maɪtə/ → [maɪrə] /saɪdə/ → [saɪrə] /laɪf/ → [laɪf]	/maɪtə/ → [maɪrə] /saɪdə/ → [saɪrə] /laɪ#fɔɪ/ → [laɪ fɔɪ]	/maɪtə/ → [maɪrə] /saɪdə/ → [saɪrə] /laɪf/ → [laɪf] /laɪ#fɔɪ/ → [laɪ fɔɪ]

Among these data sets, opaque ‘mitre-cider’ has no surface-true and surface-apparent evidence regarding the opaque process (raising). There is only indirect evidence: if

raising is not employed, there will be no [ΛI] in /maɪtə/ → [mΛɪtə]. At the same time, opaque ‘mitre-cider-life’ has evidence for the opaque process’ applying transparently, since the [f] in ‘life’ is not subject to flapping: raising must be an independent process at some stratum. Opaque ‘mitre-cider-lie-for’ provides evidence for the non-application of raising at the phrase level, since ‘lie for’ does not undergo raising despite having [f] after the diphthong across a word boundary. Finally, opaque ‘mitre-cider-life-lie-for’ combines both pieces of evidence.

Our learner had access to the four constraints in (55). As in the Southern French case, only two derivational levels were used: a word level and a phrase level.

Whenever *_{AI,əʊ}/[-voice] is sufficiently higher weighted than Ident(low), the raising process will be in effect. The flapping process is active when the weight of *_{VTV} is sufficiently higher than the weight of Ident(sonorant)²³.

(55) Constraint set for Canadian Raising simulations

- a. Ident(low) : One violation mark for raising or lowering a diphthong.
- b. Ident(sonorant) : One violation mark for any underlying consonant whose [±sonorant] specification is not identical to that of its output correspondent. This constraint penalizes the transition from /t, d/ to flap, or *vice versa*.
- c. *_{VTV} : One violation mark for an alveolar stop [t, d] in between two vowels of which the first is stressed and the second is not.
- d. *_{AI,əʊ}/[-voice] : One violation mark for a non-raised diphthong before a voiceless consonant.

The opaque interaction is captured if there is raising but no flapping at the word level, and flapping but no raising at the phrase level (see section 3.1 on the impossibility of capturing the interaction within one stratum). This is reflected in the weights found at successful learning trials for ‘mitre-cider-life-lie-for’. At the word level, *_{AI,əʊ}/[-voice]

²³ We assume that the transition from /t,d/ to [ɾ] entails a change in [±sonorant].

is far above Ident(low), and * \acute{V} TV is far below Ident(sonorant); the opposite is true of the phrase level:

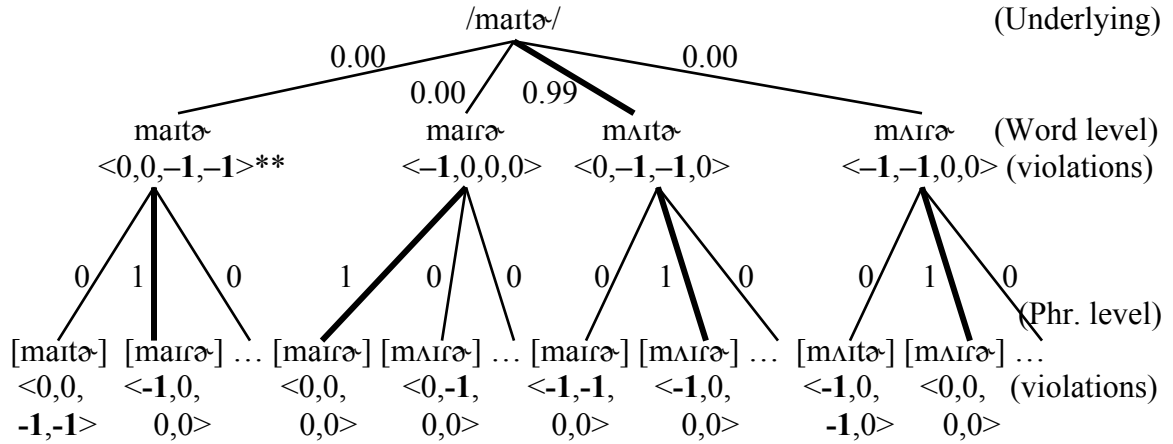
Table 20. Sample successful weights for opaque ‘mitre-cider-life-lie-for’

Word level				Phrase level			
Ident (son)	Ident (low)	* \acute{V} TV	* $a_I, a_U / _ [-vce]$	Ident (son)	Ident (low)	* \acute{V} TV	* $a_I, a_U / _ [-vce]$
10.44	5.02	0	11.13	0	6.81	6.12	0

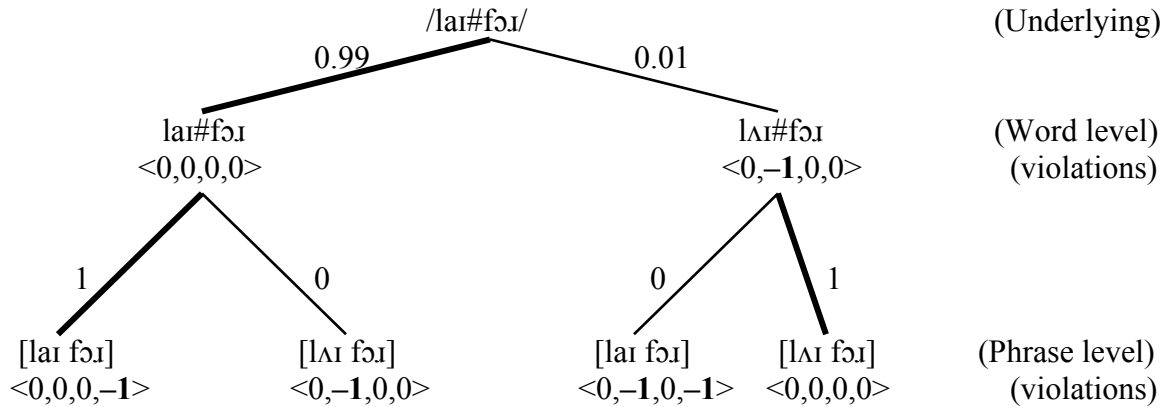
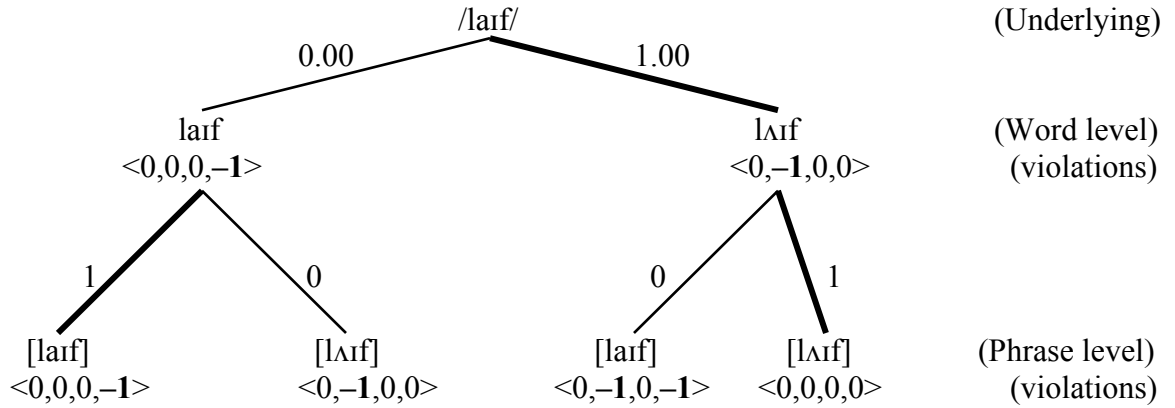
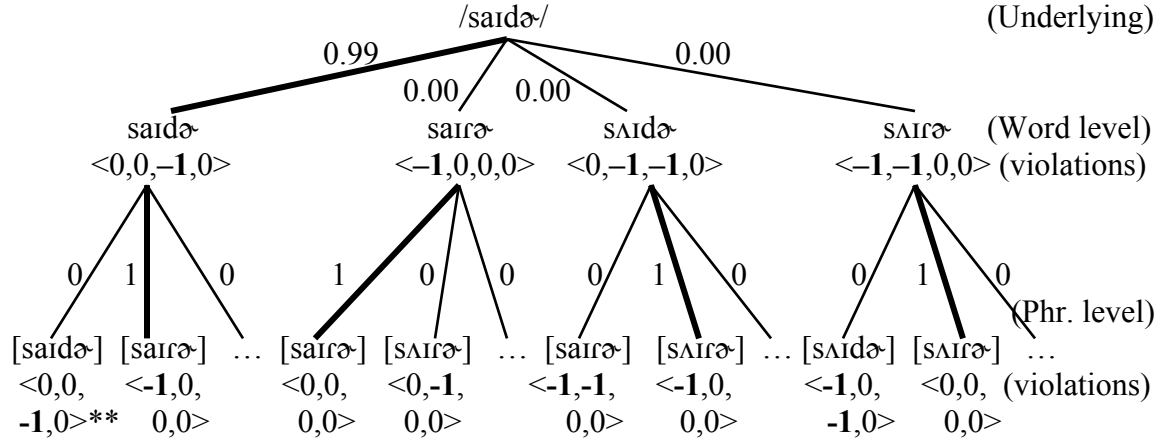
Figure 9 shows derivation graphs (see section 3.2.1) for all inputs in the ‘mitre-cider-life-lie-for’ data set. The weights used are the same ones used in Table 20. Violations are shown in the order Ident(sonorant) – Ident(low) – * \acute{V} TV – * $a_I, a_U / _ [-voice]$.

Not all phrase level candidates are shown for lack of space. The full set of candidates considered at the phrase level is the same as the candidate set at the word level²⁴.

Figure 9. Derivation graphs for opaque Canadian Raising



²⁴ Note that we do not consider mappings of the types $\ast < m\{a, \Lambda\}ɪrə \rightarrow [m\{a, \Lambda\}ɪdə] >$ or $\ast < s\{a, \Lambda\}ɪrə \rightarrow [s\{a, \Lambda\}ɪtə] >$ at the phrase level. However, we are confident that presence of these mappings would not significantly change our results. This is because these mappings have a strict superset of the violations that the fully faithful mapping has, similarly to the mappings $\ast < m\{a, \Lambda\}ɪrə \rightarrow [m\{a, \Lambda\}ɪtə] >$ and $\ast < s\{a, \Lambda\}ɪrə \rightarrow [s\{a, \Lambda\}ɪdə] >$ that were included.



**Violations given in the order <Ident(sonorant), Ident(low), *VTV, *aɪ,aʊ/[-voice]>

The resulting surface form probabilities, shown in Table 21, match the actual Canadian English data. As can be seen in Figure 9, it is essential for the generation of these

probabilities that /maɪtə/ and /laɪf/ exhibit raising at the word level while /saɪdə/ and /laɪ#fɔɪ/ do not, since the phrase level preserves diphthongs faithfully.

Table 21. Surface form probabilities generated by the graphs in Figure 9

/maɪtə/	probability	/saɪdə/	probability	/laɪf/	probability	/laɪ#fɔɪ/	probability
[maɪtə]	0.00	[saɪdə]	0.00	[laɪf]	0.00	[laɪ fɔɪ]	0.99
[maɪrə]	0.00	[saɪrə]	0.99				
[mʌɪtə]	0.00	[sʌɪdə]	0.00	[lʌɪf]	1.00	[lʌɪ fɔɪ]	0.01
[mʌɪrə]	0.99	[sʌɪrə]	0.00				

As opposed to the opaque interaction, the transparent interaction does not need to time flapping after raising. In fact, for transparent ‘mitre-cider’ and ‘mitre-cider-lie-for’, raising need not be represented in the grammar at all, because the raised diphthong is absent from the data – see the lack of weight on *aɪ,aʊ/[-voice] at either level in (28a-b).

Table 22 shows sample weights for successful runs of the transparent datasets. For transparent ‘mitre-cider-life’, raising can be represented either at the word level, by giving high weight to word level *aɪ,aʊ/[-voice] and zero weight to word level Ident(sonorant), as in line c. – or at the phrase level by giving high weight to *VTV and *aɪ,aʊ/[-voice] at the phrase level, as in line d.

Transparent ‘mitre-cider-life-lie-for’, however, requires raising at the word level, because the non-raised diphthong in /laɪ#fɔɪ/ → [laɪ fɔɪ] precludes raising at the phrase level. The weights found for this dataset are very similar to the second set of weights for ‘mitre-cider-life’, as shown in line e.

Table 22. Sample weights for successful runs of various transparent datasets

	Dataset (transparent)	Word level				Phrase level			
		Ident (son)	Ident (low)	* \check{V} TV	*aI,aU/ [-vce]	Ident (son)	Ident (low)	* \check{V} TV	*aI,aU/ [-vce]
a.	‘mitre-cider’	0	6.78	0	0	0	6.78	6.78	0
b.	‘mitre-cider-lie-for’	0	7.15	0	0	0	7.15	6.75	0
c.	‘mitre-cider-life’, var. 1	0	6.41	0	0	0	5.72	5.73	11.45
d.	‘mitre-cider-life’, var. 2	0	6.07	0	11.74	0	6.77	6.36	0
e.	‘mitre-cider-life-lie-for’	0	6.34	0	11.98	0	7.03	6.34	0

Thus, the opaque interaction of raising and flapping requires raising at the word level and flapping at the phrase level. The transparent interaction, however, only requires flapping at the word level, while raising can be represented at either level.

Finally, the addition of both ‘life’ and ‘lie for’ to the transparent data set leads to the necessity of representing raising at the word level, even though this is not required for the transparent interaction otherwise.

3.3.3 Results

Simulations were run as described in section 3.2.2. All four data sets were examined with both opaque and transparent interaction between raising and flapping. The same 100 sets of initializations drawn i.i.d. from a uniform distribution over [0,10] were used for all 8 datasets. The results are as follows:

Table 23. Results for Canadian English, for 100 runs

	Opaque	Transparent
Dataset	Learned successfully out of 100 runs	Learned successfully out of 100 runs
‘mitre-cider’	51	100
‘mitre-cider-life’	61	99
‘mitre-cider-lie-for’	87	100
‘mitre-cider-life-lie-for’	92	93

As can be seen in Table 23, independent evidence regarding the opaque process yields a

clear increase in learnability for the opaque cases. Evidence that the opaque process is word-bounded (‘lie for’) has a stronger effect than evidence for transparent activity of the opaque process (‘life’).

For all transparent datasets except ‘mitre-cider-life-lie-for’, performance is (almost) at ceiling. For ‘mitre-cider-life-lie-for’, however, the opaque and transparent versions are learned at a near equal rate.

Whenever (opaque or transparent) Canadian English is not learned successfully (i.e., at least one surface form with a probability of 1 in the learning data is given a probability of 0.5 or less by the grammar), the learner ends up in one of the 4 local optima that are summarized in Table 24. The table lists, for each underlying representation, the surface representations which have more than 0.00 probability in that local optimum. The symbol ‘~’ will be used as a shorthand for 0.50 probability on both surface representations shown, unless indicated otherwise. Weights that generate each local optimum are given in Table 25.

Table 24. Local optima found for Canadian English simulations

Optimum	Occurs in:	Inputs	Outputs
I	Opaque ‘mitre-cider’, ‘mitre-cider-lie-for’, ‘mitre-cider-life-lie-for’	/maɪtə/ /saɪdə/ (/laɪf/) (/laɪ#fɔɪ/)	[maɪɪɪə]~[maɪɪə] [saɪɪɪə]~[saɪɪə] ([laɪf]~[lɪɪf]) ([laɪ fɔɪ]~[lɪɪ fɔɪ])
II	Opaque ‘mitre-cider-lie-for’, ‘mitre-cider-life-lie-for’	/maɪtə/ /saɪdə/ (/laɪf/) /laɪ#fɔɪ/	[maɪɪɪə] 0.67 ~ [maɪɪə] [saɪɪɪə] 0.67 ~ [saɪɪə] ([lɪɪf]) [laɪ fɔɪ] 0.67 ~ [lɪɪ fɔɪ]
III	Opaque ‘mitre-cider-life’ Transparent ‘mitre-cider-life’	/maɪtə/ /saɪdə/ /laɪf/ -	[maɪɪɪə]~[maɪɪə] [saɪɪɪə]~[saɪɪə] [lɪɪf]
IV	Transparent ‘mitre-cider-life-lie-for’	/maɪtə/ /saɪdə/ /laɪf/ /laɪ#fɔɪ/	[maɪɪə] [saɪɪə] [laɪf]~[lɪɪf] [laɪ fɔɪ]~[lɪɪ fɔɪ]

Table 25. Sample weights for local optima

	Word level				Phrase level			
	Ident (son)	Ident (low)	*VTV	*a _l ,a _o / [-vce]	Ident (son)	Ident (low)	*VTV	*a _l ,a _o / [-vce]
Local optimum I	0	0	0	0	0	0	7.75	0
Local optimum II	0	6.60	0	0	0	5.91	5.92	5.90
Local optimum III	0	0.69	6.14	6.54	6.51	5.86	0.04	0
Local optimum IV	0	0	0	0	0	0	7.01	6.60

These four local optima have in common that they try to represent both processes with minimal appeal to the interaction between levels. For instance, local optima I and IV are attempts to represent the data without appealing to underlying representations, by setting all Faithfulness constraints to zero.

It is interesting that local optimum I (together with local optima II and III) has raising before underlyingly voiced /d/ in ‘cider’. This is not because raising is triggered by any constraint in that context, but because the weight of the constraints that regulate the realization of /a_l/ are either zero, or tied, so that they cannot decide between the two and produce a uniform distribution over [a_l] and [Λ_l] instead. For instance, in local optimum I, Ident(low) and *a_l,a_o/[-vce] have zero weight at both levels, so that there is no force to increase or reduce the probability of [Λ_l] relative to that of [a_l].

Local optimum III does not represent the raising process at the word level, as necessary for any data set that involves ‘life’ with a raised diphthong and ‘lie for’ with a non-raised diphthong (see section 3.3.2). This means that the learner interprets the variation between [a_l] and [Λ_l] as free variation (every word with an /a_l/ vowel has a 50% chance of that vowel’s being pronounced as [Λ_l]) instead of a situation of complementary distribution.

Finally, local optimum II is a consequence of representing both raising and

flapping at the word level. Since applying raising and flapping at the same level leads to lack of raising in ‘mitre’, variation between raising and non-raising is created by lowering the weight of word level Ident(low).

3.3.4 Summary

Summarizing, we have found that independent evidence about the opaque pattern’s stratal affiliation can significantly improve the learnability of the opaque interaction – especially the addition of evidence for non-application of the opaque process between words. Furthermore, the presence of both ‘lie’ and ‘lie for’ made the transparent interaction more difficult to learn, in fact destroying the learning advantage of the transparent interaction over the opaque one.

Whenever the languages are not learned successfully, either phrase level Faithfulness is given zero weight, making it impossible to transfer information from the word level to the phrase level, or raising and flapping are represented at the same level when they need to be represented at different levels. We will now turn to a discussion of some obstacles encountered by our learner that seem to have led some data sets to arrive at a local optimum more often than others.

3.4 Difficulties in learning hidden structure

3.4.1 Cross-level dependencies

The relative difficulty of learning opaque interactions in our framework seems to stem from the fact that the effectiveness of the weightings on each level depends on the weights on the other level. Specifically, high weight on phrase level Faithfulness is only effective when word level constraints are weighted appropriately, while the result of word level weighting can only be transmitted to the surface representation when phrase

level Faithfulness has a high enough weight.

We will show here how failure of finding appropriate weights on both word level constraints and phrase level Faithfulness simultaneously leads to local optima, and we will show that this scenario is more likely to occur in opaque cases than in transparent cases. Furthermore, we will go through the variants of the opaque Canadian English simulations, and show how the learnability differences can be accounted for in terms of their inherent bias towards local optima.

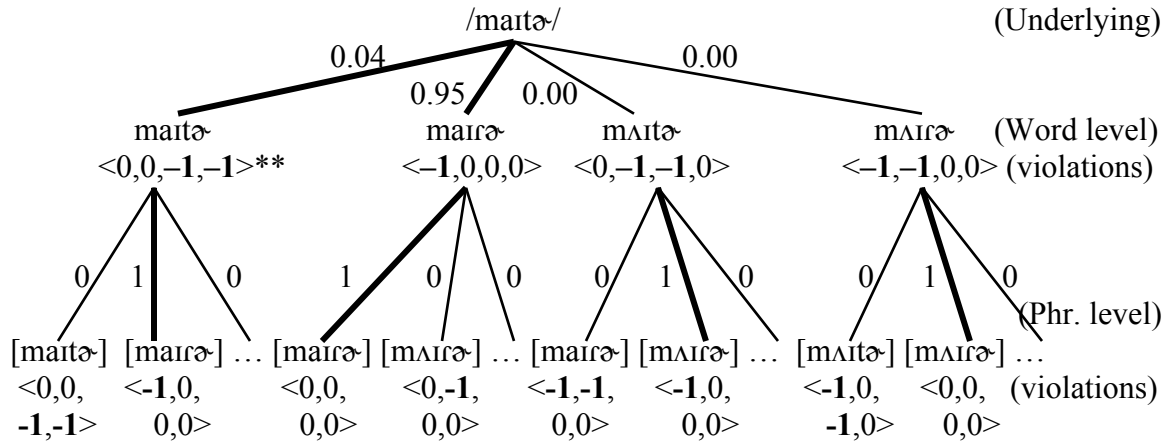
When the learner has not found weights that generate a desirable distribution at the word level, the learner gets closer to its objective by lowering the weights of phrase level Faithfulness instead of making these weights higher. For instance, consider the weighting below, which leads to a local optimum for opaque ‘mitre-cider’:

Table 26. Sample initialization that leads to local optimum for ‘mitre-cider’

Word level				Phrase level			
Ident(son)	Ident(lo)	*VTV	*aɪ,aʊ/ [-vce]	Ident(son)	Ident(lo)	*VTV	*aɪ,aʊ/ [-vce]
1	7	3	1	0	6.28	6.29	0

At the word level, this weighting gives a non-raised diphthong in ‘mitre’ highest probability, because high-weighted *VTV and low-weighted leads to flapping which blocks diphthong raising. At the same time, it has flapping and lack of raising on the phrase level, as desired for the opaque interaction (cf. section 3.3.2)

Figure 10. Derivation graph ('mitre' only) for weights in Table 26



**Violations given in the order <Ident(sonorant), Ident(low), *VTV, *ai,au/[-voice]>

In cases like this, KL-divergence decreases (and fit to the data increases) as the weight of phrase level Ident(low) at the phrase level goes to zero. Note that maximizing fit (which, in this case, is defined as minimizing KL-divergence) is not the same as maximizing the sum of the winners' probabilities. As can be seen in Table 28, the winners' probabilities (which have been highlighted for convenience) sum to 1.0 in each case. However, since the formula for KL-divergence (see (49) and (50) in section 3.2.2) takes the logarithm of every winner's probability before summing them, every individual probability's deviance from 1 has a separate influence on the resulting measure of fit.

Table 27. Observed distribution for opaque 'mitre' and 'cider'

/maɪtə/				/saɪdə/			
[maɪtə]	[maɪrə]	[mʌɪtə]	[mʌɪrə]	[saɪdə]	[saɪrə]	[saɪdə]	[saɪrə]
0	0	0	1	0	1	0	0

Table 28. KL-divergence for grammars with varying weights of phrase level Ident(low) and all other weights as in Table 26

		High phrase level Faith	Lower phrase level Faith	Lowest phrase level Faith
Weights (word level)	Ident(son)	1	1	1
	Ident(low)	7	7	7
	*VTV	3	3	3
	*aɪ/ [-vce]	1	1	1
Weights (phrase level)	Ident(son)	0	0	0
	Ident(low)	6.28	1	0
	*VTV	6.29	6.29	6.29
	*aɪ/ [-vce]	0	0	0
/maɪtə/: SR probabilities	[maɪtə]	0	0	0
	[maɪrə]	1	0.72	0.5
	[mʌɪtə]	0	0	0
	[mʌɪrə]	0	0.27	0.5
/saɪdə/: SR probabilities	[saɪdə]	0	0	0
	[saɪrə]	1	0.72	0.5
	[sʌɪdə]	0	0	0
	[sʌɪrə]	0	0.27	0.5
KL-divergence		5.87	1.63	1.39

If the learner has assigned zero weight to phrase level Faithfulness, moving towards appropriate weights to word level constraints does not lower KL-divergence. This is illustrated in Table 29 below. Word level and phrase level expected probabilities are shown for ‘mitre’ only, but KL-divergence is computed for both ‘mitre’ and ‘cider’. Phrase level constraint weights are as given in Table 26, except for Ident(low), whose weight is set to zero.

Table 29. KL-divergence for ‘mitre-cider’ when phrase level Ident(low) has zero weight

		Worst word level weights			Best word level weights
Weights (word level)	Ident(son)	1	4	6	7
	Ident(low)	7	3	3	3
	*VTV	3	2	1	0
	*aɪ/ [-vce]	1	4	6	7
Weights (phrase level)	Ident(son)	0	0	0	0
	Ident(low)	0	0	0	0
	*VTV	6.29	6.29	6.29	6.29
	*aɪ/ [-vce]	0	0	0	0
/maɪtə/: word level output probabilities	maɪtə	0.05	0.09	0.04	0.02
	maɪrə	0.95	0.64	0.11	0.02
	mɪtə	0	0.23	0.84	0.96
	mɪrə	0	0.03	0.01	0
/maɪtə/: SR probabilities	[maɪtə]	0	0	0	0
	[maɪrə]	0.5	0.5	0.5	0.5
	[mɪtə]	0	0	0	0
	[mɪrə]	0.5	0.5	0.5	0.5
KL-divergence		1.39	1.39	1.39	1.39

As can be seen in the table above, raising the weights of word level Ident(low) and *VTV and lowering those of word level Ident(sonorant) and *aɪ,aʊ/ [-voice] leads to a desirable result at the word level: raising and no flapping (cf. Figure 9 in section 3.3.2). However, this information is not factored into the distribution over UR/SR pairings if phrase level Ident(low) has a weight of 0.

However, phrase level Ident(low) has a motivation to decrease its weight until word level outputs with a raised diphthong gain a cumulative probability of at least 0.5. Setting Ident(low) to zero means that the phrase level will consider raised diphthongs and non-raised ones with equal probability. Thus, the fit to the observed data will only improve when the word level provides more than 0.5 cumulative probability for the raised diphthong in ‘mitre’.

Taken together, this means that if there is a sufficient distance between the

current word level weights and a set of desirable word level weights, phrase level Faithfulness might be set to zero before desirable word level weights might be found. We will call this the bottleneck effect: word level information needs to travel through phrase level Faithfulness constraints in order to have an effect on the UR-to-SR distribution.

3.4.2 Advantage from evidence for stratal affiliation

As we showed in section 3.3.3, the addition of various kinds of evidence regarding the opaque process (raising) dramatically increases the likelihood that the opaque interaction will be learned.

Adding ‘life’ to the opaque interaction means that raising at either the word level or the phrase level will be rewarded, regardless of the weighting of the flapping constraint * \acute{V} TV. This means that increasing the weight of word level * a_I, a_U /[-voice] leads to a sharper drop in KL-divergence for opaque ‘mitre-cider-life’ than for opaque ‘mitre-cider-lie-for’²⁵.

²⁵ However, lowering the weight of phrase level Ident(low) also leads to a stronger decrease in KL-divergence compared to opaque ‘mitre-cider’, so that the bottleneck effect becomes stronger with the addition of ‘life’. This is the likely cause of the modest effect of ‘life’ on learnability.

Table 30. Adding ‘life’ leads to a stronger effect of representing raising at the word level

		Lowest position of word level *aɪ,aʊ/[-vce]		Highest position of word level *aɪ,aʊ/[- vce]
Weights (word level)	Ident(son)	1	1	1
	Ident(low)	7	7	7
	*ŴTV	3	3	3
	*aɪ/ [-vce]	1	7	14
Weights (phrase level)	Ident(son)	0	0	0
	Ident(low)	6.28	6.28	6.28
	*ŴTV	6.29	6.29	6.29
	*aɪ/ [-vce]	0	0	0
KL-divergence for ‘mitre-cider’		5.87	5.85	5.85
KL-divergence for ‘mitre-cider-life’		11.31	6.54	5.85

Adding ‘lie for’ to the opaque interaction means that high weight on phrase level Ident(low) is penalized less strongly when the word level weights do not generate the desirable candidates with high probability²⁶. This is because word ‘lie’ itself contains neither a flapping nor a raising context, so that flapping and raising constraints do not interact with the identity of the diphthong at the word level. It is not necessary to determine the mutual weighting of word level *ŴTV and Ident(son) or *aɪ,aʊ/[-voice] and Ident(low) to give the desirable word level output for ‘lie for’ more than 0.5 probability – non-zero weight on word level Ident(son) is sufficient.

²⁶ Another effect of adding ‘lie for’ is that high weight on phrase level *aɪ,aʊ/[-voice] leads to a much stronger increase in KL-divergence, because this constraint prefers a raised diphthong in ‘lie for’.

Table 31. Adding ‘lie for’ makes it less attractive to lower weight on Ident(low)

		Highest position of phrase level Ident(low)		Lowest position of phrase level Ident(low)
Weights (word level)	Ident(son)	1	1	1
	Ident(low)	7	7	7
	*√TV	3	3	3
	*aI/ [-vce]	1	1	1
Weights (phrase level)	Ident(son)	0	0	0
	Ident(low)	6.28	1	0
	*√TV	6.29	6.29	6.29
	*aI/ [-vce]	0	0	0
KL-divergence for ‘mitre-cider’		5.87	1.63	1.39
KL-divergence for ‘mitre-cider-lie-for’		5.87	1.94	2.08

When both ‘life’ and ‘lie for’ are present in the data set, the grammar must represent the raising process at the word level in order to generate lack of raising in ‘lie for’ (because word level *aI,aU/ [-voice] cannot see that ‘lie for’ has /aI/ before /f/, while phrase level *aI,aU/ [-voice] can). As shown in section 3.3.2, this introduces an additional dependency between word and phrase level for the transparent datasets: while the transparent interaction between flapping and raising does not require raising to apply before flapping, the combination of ‘life’ and ‘lie for’ does.

This creates a bottleneck effect for the transparent dataset, which explains the decrease in learnability for transparent ‘mitre-cider-life-lie-for’. The increase in learnability for opaque ‘mitre-cider-life-lie-for’, on the other hand, can be seen as the cumulative effect of ‘life’ and ‘lie for’ on the opaque interaction, as reviewed above.

3.5. Concluding remarks

We have presented here an approach to learning opaque and transparent interactions in a weighted constraint version of Stratal OT. Our goal was to investigate

whether the general setup of Stratal OT – chained parallel OT grammars with independent rankings or weightings of constraints – predicts learnability differences between opaque and transparent process interactions, despite the fact that this setup does not have a built-in bias in favor of transparent interactions. In particular, we sought to test Kiparsky’s (2000) hypothesis that evidence of a process’ stratal affiliation makes it easier to learn opaque interactions in a Stratal setup.

Our first case study was opaque tensing/laxing in Southern French. We found that it was learned at a high rate of accuracy, but the solution space has a local optimum – one where the grammar does not represent the phonological process at all, which yields free variation in the data.

We then looked at the opaque interaction between diphthong raising and flapping in Canadian English (Joos 1942, Idsardi 2000 and references therein). The opaque raising process also applies in contexts where flapping is irrelevant, and it does not apply across word boundaries – both of which constitute evidence for the stratal affiliation of raising. Furthermore, the opaque interaction in Canadian English has a clear transparent counterpart, as described by Joos (1942).

We found that, without additional evidence for stratal affiliation, the opaque interaction was learned at a rate of about 50%, while its transparent counterpart was learned at ceiling. However, addition of this additional evidence improved the learnability of the opaque interaction to a maximum of 92%, while the learnability of the transparent interaction descended to a similar rate. This confirms Kiparsky’s prediction: evidence of the stratal affiliation of raising does improve its learnability when it is opaque.

Our explanation for this effect has to do with a bottleneck effect in the transmission of information from earlier derivational levels through Faithfulness constraints. This effect makes it difficult to find the global optimum when initializations need a large amount of adjustment before they can generate the desirable distribution over word level output forms. However, it is mitigated for opaque Canadian English by information about an opaque process' stratal affiliation, because this information either boosts desirable word level weightings, punishes undesirable phrase level weightings, or diminishes the bottleneck effect in general. See section 3.4.2 for details.

We limited ourselves here to two case studies and one learning implementation. Other approaches to learning Stratal OT are possible too: ranked constraint learning with Maximum Likelihood (Jarosz 2006a), (Noisy) Harmonic Grammar (Coetzee 2009a, Boersma and Pater 2016), Stochastic OT (Boersma 1998). These might differ in their particular learning strategies, and the distribution over outputs generated at local optima. Moreover, the current approach's performance should be compared to that of the parallel multi-level learner developed in Boersma and van Leussen (2015).

The mechanisms responsible for the learnability differences that we found are quite general. Interdependence between phrase level Faithfulness and appropriate weighting at the word level poses a challenge for finding a grammar that generates the learning data. For this reason, we predict that other learning approaches will find results similar to ours. However, this prediction needs testing in future work that compares the learnability of these and other opaque interactions in this Stratal framework and in other learning frameworks.

Future work is also needed to examine how other cases of opacity behave in our

framework. In particular, more complex interactions which involve more constraints and/or more derivational levels would be essential to test our predictions. More complex versions of the case studies used here should be considered as well (for instance, a variant of the French case study in which the weight of syllabification constraints must be learned). In general, however, our work agrees with Jarosz's (2015) findings that opaque cases are expected to be learned less easily than transparent ones.

CHAPTER 4

LEARNING THE DIVIDE BETWEEN RULE AND EXCEPTIONS FOR DUTCH STRESS

4.1 Introduction

In this chapter, I propose a novel method of inducing word-specific (or morpheme-specific) phonological generalizations. In particular, I propose a novel learner for lexically indexed constraints (Kraska-Szlenk 1995, Pater 2000, 2010), making it possible to induce such constraints in probabilistic OT, as opposed to previous learners, which only work with categorical OT learners (Becker 2009, Coetzee 2009b, Pater 2010).

Lexically Indexed Constraint Theory, which will be introduced in section 4.1.1, creates exceptional constraint rankings for certain lexical items by inserting word-specific copies of constraints into the grammar. Alternatives to this theory include Cophonology Theory (Itô and Mester 1995, Orgun 1996, Inkelas 1998, Inkelas and Orgun 1995, 1998, and Inkelas and Zoll 2007) and Sublexical Phonology (Linzen, Kasyanenko, and Gouskova 2013, Gouskova, Newlin-Lukowicz, and Kasyanenko 2015, Becker and Gouskova 2016). The main difference between these theories and Lexically Indexed Constraint Theory is that the former assume separate grammars for different words, while the latter does not. The learner here focuses on finding a single grammar that accounts for both rule-obeying forms and exceptions, but it could, in principle, be adapted to Cophonology Theory. See Allen and Becker (under revision) for an existing learning for Sublexical Grammar Theory.

The learner presented here aims to find a division of the lexicon into rule-obeying and exceptional forms. As argued in chapter 1, this is a hidden structure learning

problem. There is no plausible way in which a language-acquiring child can receive information about a word's (or morpheme's) exceptional status directly from the signal. At the same time, if the grammar is to generate the correct surface form for exceptional words, it needs to mark these words as exceptional. This means that exception marking is a type of hidden structure.

In the goal of trying to mark exceptional forms and leave rule-obeying forms unmarked, the current approach differs from some other approaches to learning lexically indexed constraints such as Becker (2007, 2008, 2009) and Becker and Fainleib (2009), who attempt to build speakers' stochastic judgments about non-words into the grammar itself by splitting constraints into several versions, each indexed to a separate group of words, and do not distinguish between 'exceptional' and 'non-exceptional' words.

While such innovations are motivated by behavioral data and mirror similar proposals designed to include within-word variation in OT grammars (Anttila 1997, Coetzee and Pater 2011), they diverge from the traditional and perhaps more idealistic view that speakers have a single unmarked grammar for a language, and forms that contradict the patterns laid down by this grammar are somehow more marked than words that follow these patterns. For instance, in Chomsky and Halle (1968), there is one single set of rules for all words, but words can be marked underlyingly for not undergoing a particular rule; see Wolf (2011) for an overview of other literature that develops this view. In particular, all existing theoretical analyses of Dutch stress (e.g., van der Hulst 1984, Kager 1989, Nouveau 1994, van Oostendorp 1997, 2012, 2014), which is a case study that will be used to test the learner developed here (see below and section 4.2), are built on the assumption of an unmarked default pattern and marked exceptions.

Furthermore, there is evidence that children make errors in the direction of the regular pattern (Wonnacott and Newport 2005, Hudson Kam and Newport 2005, Newport 2016), which is an effect that Nouveau (1994) has also documented for Dutch stress. Albright and Hayes (2003) find a similar effect in English-speaking adults who were asked to produce past tenses for nonce verbs but were reluctant to produce any irregular past tense forms at all, even for verbs that met the structural description of one of the irregular past tense patterns in English.

The work presented here is an attempt to see whether grammars that work under the assumption of an unmarked default and marked exceptions can nevertheless be learned in the face of widespread exceptionality, which Dutch stress provides. See sections 4.2.1.2 and 4.2.2.2 for more details on the exceptionality present in Dutch stress.

The learner itself is set up in the framework of Expectation Driven Learning (Jarosz submitted), which will be elaborated in section 4.3. This learning framework, which is based in Expectation Maximization (Dempster, Laird, and Rubin 1977), allows to build a learner for lexically indexed constraints in a probabilistic setting. Previous work on learning lexically indexed constraints (Becker 2007, 2009, Coetzee 2009b, and Pater 2010) has been done within the framework of Recursive Constraint Demotion (Tesar 1995), which is a categorical framework: there is no point at which several constraint rankings are entertained at the same time with a probability (other than 0 or 1) attached to each ranking, and there is no sense in which different outputs can win for the same input with a probability (other than 0 or 1) attached to each output.

Jarosz (2013a) shows that hidden structure problems (metrical structure is used as a case study) benefit from a probabilistic approach. A categorical learning approach (for

metrical structure, this is Robust Interpretive Parsing, Tesar 1998) cannot entertain several hypotheses at the same time, but must choose a single hypothesis, even if turns out to be wrong. At the same time, a probabilistic learning approach can entertain several hypotheses, each with a certain probability, and through this, not commit to a hypothesis regarding hidden structure until there is sufficient support for it from the data. See also Manning and Schütze (1999) for broader arguments for the effectiveness of probabilistic approaches for learning language.

At the same time, probabilistic OT allows for the representation of within-word variation (Anttila 1997, Coetzee and Pater 2011). By allowing several outputs to win for the same input, each with a certain nonzero probability, the occurrence of different phonological forms of the same word can be modeled. Thus, if a model of finding exceptions (accounting for between-word variation) is to be consistent with learning other types of hidden structure efficiently, as well as with the reality of within-word variation, a probabilistic framework is highly desirable. The current proposal allows such integration.

The essence of the learning proposal is that, for each pair that can be constructed of the constraints given to the learner, the learner compares the lexicon-wide preference for that constraint, computed according to Jarosz's (submitted) proposal, to the word-specific preference for that constraint. Given a constraint pair A and B, a word is considered to be deviant if a word's preference for the ranking $B \gg A$ and the lexicon-wide preference for the ranking $A \gg B$ both exceed a certain threshold. If a constraint pair has the highest sum of deviance for these deviant items at the current iteration of the learning algorithm, this constraint pair is used to induce or update an existing indexed

constraint – see section 4.3.3 for a detailed description of how this is done.

The algorithm proposed here will be applied to Dutch main stress (van der Hulst 1984, Kager 1989, Nouveau 1994, van Oostendorp 1997, 2012, Gussenhoven 2014). This stress system, along with English and German (see Domahs et al. 2014 for an overview), is notable for having a complex Quantity-Sensitive (QS) default rule, as well as widespread exceptionality: main stress is attested on the antepenultimate, penultimate, and final syllable of any combination of Heavy and Light in the last two syllables (see section 4.2.2).

This exceptionality, for instance, makes the data consistent with a Quantity-Insensitive (QI) analysis, if all non-penultimate stress data are considered exceptions (van Oostendorp 2012). Nevertheless, only the QS analysis is supported by psycholinguistic evidence (Nouveau 1994, Ernestus and Neijt 2008, Domahs et al. 2014; see section 4.2.3 for more details).

The ambiguity of splitting these Dutch data into rule-obeying and exceptional words makes it an ideal test case for the algorithm presented here. Regular stress rules, like the Dutch QS default stress rule, are phonotactic in nature, since they can be represented without recourse to Faithfulness constraints. Ideally, the learner would find a phonotactic grammar that corresponds to the QS default rule given by van der Hulst (1984), Kager (1989), and others. At the same time, the learner should be successful at marking all and only the exceptional forms as exceptions for some constraint ranking.

As I will show in section 4.4.3 of this chapter, the learner does indeed find a QS grammar that makes correct predictions on novel items, and it accounts for all exceptions to the rule expressed by this QS grammar. The mechanism for expressing exceptionality

used by the learner is the indexation of constraints to particular lexical items. The details of the theory of lexically indexed constraints used here are given in section 4.1.1 below.

4.1.1 Lexically Indexed Constraint Theory

Lexically Indexed Constraint Theory (Kraska-Szlenk 1995, Pater 2000; see also Itô and Mester 1999 for a very similar proposal for lexical strata) proposes that word-specific effects in phonology arise from constraints' being indexed to particular words or morphemes. The indexation of a constraint to a set of words/morphemes means that the constraint only has violations for those. For instance, a constraint Max_i which is indexed to the words that have index i will only have violations for those words:

(56)

Max_i : One violation for every input segment without an output correspondent, but only if the input segment is in an input that has the index i .

inputs = {/tat/, /pat_i/, /sat/, ...}

Indexed constraints make it possible to postulate exceptions to a lexicon-wide ranking $A \gg B$. For instance, suppose that there is a lexicon-wide ranking $\text{NoCoda} \gg \text{Max}$, which leads to deletion of all codas, as in (57a). In this case, an indexed constraint Max_i can be ranked above NoCoda , so that all exceptional words with index i are allowed to have codas, as shown in (57b). However, non-exceptional words as well as novel words are still disallowed from having codas, since they do not have the index i and thus do not have violations of Max_i . This is illustrated in (57c).

(57) An illustration of the effects of lexically indexed constraints

a. $\text{NoCoda} \gg \text{Max}$: no codas on the surface – potential codas deleted

/tat/	NoCoda	Max
tat	*!	
☞ ta		*

b. $\text{Max}_i \gg \text{NoCoda} \gg \text{Max}$: codas allowed in words with the index i

/pat _i /	Max_i	NoCoda	Max
☞ pat		*	
pa	*!		*

c. $\text{Max}_i \gg \text{NoCoda} \gg \text{Max}$: codas disallowed in words without the index i

/sat _i /	Max_i	NoCoda	Max
sat		*!	
☞ sa			*

Becker (2007, 2008, 2009), and Becker and Fainleib (2009) propose a version of Lexically Indexed Constraint Theory in which there is no necessity of a general-to-specific relation between the constraint that applies to exceptional forms and the constraint that applies to non-exceptional forms. Both constraints simply receive an index, and both exceptional and non-exceptional forms are marked. This is illustrated in the tableaux in (58) below.

(58) Constraint indexation without the assumption of a lexicon-wide constraint

a. $\text{Max}_i \gg \text{NoCoda} \gg \text{Max}_j$: codas allowed in exceptional words with the index i

/pat _i /	Max_i	NoCoda	Max_j
☞ pat		*	
pa	*!		*

b. $\text{Max}_i \gg \text{NoCoda} \gg \text{Max}_j$: no codas in non-exceptional words, indexed with j

/sat _j /	Max_i	NoCoda	Max_j
sat		*!	
☞ sa			*

However, I share with Pater (2000, 2010) the assumption that an indexed constraint, such as Max_i , presupposes the existence of a lexicon-wide constraint of the same name, such as Max. Thus, every indexed constraint is in a stringency relation with respect to the constraint that is obeyed by non-exceptional forms.

This stringency relation reflects the concept of an unmarked, lexicon-wide grammar (see references at the beginning of 4.1). If all indexed constraints are removed from a grammar in the style assumed here, or if there are no words that are marked with

these indices, the grammar will be completely regular (all words will be treated according to the majority pattern). On the other hand, if all indexed constraints are removed from a grammar in the style advocated by Becker (2007, 2008, 2009), and Becker and Fainleib (2009), the grammar will be impoverished, because certain constraints will be completely absent from the grammar – for instance, the mini-grammars in (58) will not have the constraint Max.

The learner proposed here could be reformulated without the stringency assumption, using Becker's (2007, 2008, 2009), and Becker and Fainleib's (2009) assumptions. The learner proposed here (see section 4.3) assumes that indexed constraints must be induced when there is a sufficient amount of inconsistency in the mutual ranking of two constraints, let us call these A and B. When such a situation arises, the reformulated learner would simply split one of the constraints A or B into two versions: one version, for instance, A_i , for exceptional forms, and one version, for instance, A_j , for non-exceptional forms.

In its current form, however, the learner retains the stringency assumption: indexed constraints have a strict subset of the violations of their unindexed counterparts, which are predicated over the entire lexicon. Under this assumption, this means that, because a more specific constraint can generally only have any effect in a Paninian, specific-over-general ranking (although see Prince 1997 for special cases of interactions in which general-over-specific ranking can have an effect), the indexed constraint representing that idiosyncratic behavior must be above, not below its corresponding lexicon-wide constraint in order for an exceptional form to behave differently from the rest of the lexicon.

However, if coda retention was the norm and coda deletion was exceptional, then this could not be modeled by lowering Max_i below NoCoda if the non-indexed ranking is $\text{Max} \gg \text{NoCoda}$, as illustrated in (59b). Since Max is a lexicon-wide constraint, it rules out deletion for all words, including ones indexed to all lower-ranked constraints.

(59) The importance of specific-general relations in the indexed constraint framework used here

a. $\text{Max}_i \gg \text{NoCoda} \gg \text{Max}$ yields exceptional coda retention:

/pat _i /	Max_i	NoCoda	Max
☞ pat		*	
pa	*!		*

b. $\text{Max} \gg \text{NoCoda} \gg \text{Max}_i$ does not yield exceptional coda deletion

/tat _i /	Max	NoCoda	Max_i
tat		*	
☹ ta	*!		*

For the purpose of the learner proposed here, I sharpen this assumption of stringency relations between lexicon-wide and indexed constraints. I assume that any two indexed constraints derived from the same constraint must be in a general-specific relation.

This means that constraint indexation proceeds in a nested fashion. For example, Max may have the indexed version Max_i , with *i* on the inputs /pat/, /nat/, /dad/. However, a second indexed version of Max is only possible if it is derived from Max_i itself: if some subset of the words indexed to Max_i behave exceptionally, they may be indexed to a recursively indexed constraint $\text{Max}_{i,j}$.

Suppose that the toy language illustrated in the tableaux below also has a high-ranked constraint *CCC. Suppose also that the language has a suffix [-sta], but standardly disallows three-consonant clusters, even in most words which do allow codas. For instance, /pat-sta/ becomes [pasta], so that *CCC must be ranked above Max_i . However, some exceptions are allowed, including the input /nat/, in which the last consonant of the

stem is allowed to surface even with the suffix [-sta]: /nat-sta/ → [natsta].

In order to encode the fact that [natsta] defies the high-ranked constraint *CCC, it must be indexed to some version of Max that is above Max_i. Given the assumption of nested indexation, this must be a recursively indexed constraint Max_{i,j}. The input /nat/ must be indexed to both Max_i and Max_{i,j}. Tableaux for /pat-sta/ and /nat-sta/ are given below.

(60) Recursive indexation of Max: Max_{i,j} >> *CCC >> Max_i >> NoCoda >> Max

a. Once-indexed stem /pat_i/

/pat _i -sta/	Max _{i,j}	*CCC	Max _i	NoCoda	Max
patsta		*!		*	
☞ pasta			*		*

b. Twice-indexed stem /nat_{i,j}/

/nat _{i,j} -sta/	Max _{i,j}	*CCC	Max _i	NoCoda	Max
☞ natsta		*		*	
nasta	*!		*		*

Crucially, the sharpened assumption of stringency disallows the induction of a constraint Max_j alongside Max_i. In other words, words are split between rule-obeying and exceptional for every constraint, but there are never several categories of exceptions beside one another. Stratified exceptionality, as illustrated in (60), can be achieved by nested indexation.

Constraint indexation is set up in this recursive way for practical considerations. This setup maximizes generalization across exceptional words, since all word forms that need an exceptional form of a certain constraint, e.g., Max, must be indexed to the same indexed version of that constraint, e.g. Max_i. If certain exceptional words need another version of the indexed constraint, this is done by making a recursively indexed version of the same constraint, e.g., Max_{i,j}, as illustrated in (60b) above. This “lumping” of exceptional forms into one single constraint whenever possible minimizes the total

number of constraints in the learner (which affects learning time).

Other ways of setting up indexed constraints are also possible: for instance, separate indexed constraints can be set up for every word, which are later lumped into constraint indexed to groups of words. This alternative method should be explored in future work. In fact, a similar method has been employed by Moore-Cantwell and Pater (to appear). This work does not address the induction of indices, but, rather, simply assumes a separate index for each word. However, it is shown that, in such an approach, the strength of the default pattern can be derived from the relative amount of exceptions to that pattern, which is a very interesting result whose intersection with the current approach should be explored in the future.

4.1.2 Other approaches to exceptionality

Besides Lexically Indexed Constraint Theory, there are other approaches to exceptionality in OT. One of the most prominent alternatives is Cophonology Theory (Itô and Mester 1995, Orgun 1996, Inkelas 1998, Inkelas and Orgun 1995, 1998, and Inkelas and Zoll 2007), in which separate OT grammars are predicated of different words and of different morphological . The simple example of exceptional coda retention introduced above would be analyzed such that non-exceptional words are evaluated with the grammar NoCoda >> Max, whereas exceptional words are marked for being evaluated with the alternative grammar Max >> NoCoda. See Pater (2010) for a comparison between Lexically Indexed Constraint Theory and Cophonology Theory.

Another related approach is Sublexical Phonology (Linzen, Kasyanenko, and Gouskova 2013, Gouskova, Newlin-Łukowicz, and Kasyanenko 2015, Becker and Gouskova 2016, Allen and Becker under revision), in which different parts of the lexicon

have different constraint weightings, and new words are assigned to these constraint weightings by a “gatekeeper grammar”.

The relevant aspect that unites these approaches and indexed constraint theory is that exceptional forms have rankings that diverge from those in the grammar that applies to non-exceptional forms. The learner presented here is designed to find preferences for opposite rankings between the entire lexicon and individual words, but the way in which these contradictory rankings are implemented can be varied according to the framework.

4.1.3 Chapter overview

The rest of this chapter will be structured as follows. Section 4.2 will lay out the Dutch primary stress data, and will briefly sketch Nouveau’s (1994) analysis of these facts, which I will use as the basis for my simulations. Section 4.3 will introduce the learner that I used for the simulations – both the Expectation Driven Learning framework (Jarosz submitted) that was used, and the mechanism that I propose for inducing indexed constraints. Section 4.4 will then detail the setup of the simulations that were run, and the results obtained from these simulations. Finally, section 4.5 will summarize and conclude.

4.2 Dutch primary stress assignment

4.2.1 The generalizations

4.2.1.1 Non-exceptional stress

The data presented to the learner are a simplified form of Dutch primary stress data taken from Ernestus and Neijt (2008). Dutch primary stress is conventionally analyzed as a Quantity-Sensitive (QS) system with exceptions (van der Hulst 1984, Kager 1989, Nouveau 1994, van Oostendorp 1997 and others).

The standard view of the default rule (van der Hulst 1984, Kager 1989, Trommelen and Zonneveld 1989, Nouveau 1994, van Oostendorp 1997) can be summarized as follows. CVC syllables are heavy, CV syllables are light. The default location of primary stress is the penult, as in (61a). However, when the final syllable is heavy, stress shifts onto the antepenult, as in (61b), unless the penult is heavy, as in (61c), or there is no antepenult, as in (61d). The standard analysis of this rule will be introduced in section 4.2.2.

(61) Examples of non-exceptional stress in Dutch

a. penult stress: default

pi. 'ja.ma “pajamas”

a. 'χɛn.da “agenda”

,ma.ka. 'ro.ni “macaroni”

,pro.pa. 'χan.da “propaganda”

b. antepenult stress: when last syllable is heavy (=closed) and penult is light (=open)

'y.ni. sef “Unicef”

'al.ma. ,nak “almanac”

je. 'ry.za. ,lɛm “Jerusalem”

c. no antepenult stress when last syllable and penult are both heavy

χi. 'bral.tar “Gibraltar (toponym)”

e. 'lɛk.trɔn “electron”

,ro.do. 'dɛn.drɔn “rhododendron”

d. penult stress when there is no antepenult

'ro.bɔt “robot”

Word-finally, so-called superheavy syllables can occur, which either have a coda of more than one consonant (CVCC), or violate the restriction against co-occurrence of a tense vowel or diphthong and a coda in the same syllable ($CV_{\text{tense}}C$, CVJC). These syllables attract main stress to the final syllable by default, as shown in (62).

(62) Examples of (non-exceptional) final stress in words ending in a superheavy syllable

,pre.zi. 'dɛnt “president” (CVCC syllable)

,a.bri. 'kos “apricot” ($CV_{\text{tense}}C$ syllable)

,as.tro. 'naut “astronaut” (CVJC syllable)

Because of their limited distribution, superheavy syllables have been analyzed in the literature (Zonneveld 1993) as being larger than one syllable: the consonants that come after the first coda consonant or after the tense vowel are extrasyllabic, as in (63a), possibly forming its own (light) syllable, as in (63b). Under this analysis, stress on a final superheavy syllable is really an instance of default penult stress. This is also what Nouveau (1994), whose analysis I will take as the basis for my simulations, assumes. However, since this analysis is not uncontroversial, I will exclude superheavy syllables from consideration for the remainder of this chapter.

(63) Possible alternative representations of superheavies

- a. ,pre.zi.'dɛn.t
- b. ,pre.zi.'dɛn.tV

Words with schwa syllables will also be left out of consideration, since schwa syllables may never be stressed (Kager 1989, van Oostendorp 1995), and therefore influence the stress pattern. In effect, almost all words relevant to the metrical stress assignment rules are non-native (see also Ernestus and Neijt 2008), since native words that are not compounds often have schwa in unstressed position.

4.2.1.2 Exceptions

The system as illustrated in (61) is subject to widespread exceptionality, as illustrated in (64):

(64) Exceptional stress in Dutch

- a. words that end in Light-Heavy with penult stress:
 - ,ka.ta.'ma.ran “catamaran”
 - se.'le.bɛs “Celebes (toponym)”
- b. words that do not end in Light-Heavy with antepenult stress:
 - 'pa.ɣi.,na “page”
 - a.'me.ri.,ka “America”

- c. words with final stress:
 ʃo.ko.'la “chocolate”
 ʔr.χi.'de “orchid”
 kro.ko.'dɪl “crocodile”

Table 32 below shows how the weight of the last two syllables correlates with the location of main stress. These numbers are derived from all 3- and 4-syllable monomorphemic words in the Dutch section of the CELEX corpus (Baayen, Piepenbrock, and Gulikers 1995), as reported in Ernestus and Neijt (2008). These do not include words with schwa syllables or superheavy syllables.

Table 32. Number of Dutch monomorphemic words for the relevant stress-and-syllable-weight combinations, according to Ernestus and Neijt (2008); numbers shown in gray are considered to be negligibly small (smaller than 5)

Weight pattern	No. of words			Weight pattern	No. of words		
	Antep. stress	Penult stress	Final stress		Antep. stress	Penult stress	Final stress
X X L L	18	21	2	X L L	63	74	24
X X L H	1	8	5	X L H	54	7	34
X X H L	0	12	0	X H L	2	41	6
X X H H	0	1	0	X H H	2	2	2

The fact that there is so much exceptionality makes it possible to propose a different division between rule and exceptions. Van Oostendorp (2012) proposes that Dutch synchronically is a Quantity-Insensitive (QI) language: the grammar always assigns penultimate stress, while all other main stress patterns are lexically specified. Whatever correlations there are between syllable weight and main stress placement are a product of historic accident (van Oostendorp cites the fact that quantity-sensitivity only ever plays a role in the Romance stratum).

(65) QI analysis of Dutch stress (van Oostendorp 2012)

a. rule-obeying: penult stress

pi.'ja.ma “pajamas”

a.'ʒɛn.da “agenda”

ka.ta.'ma.ran “catamaran” (exceptional under standard analysis)

se.'le.bes “Celebes (toponym)” (exceptional under standard analysis)

b. exceptional: final and antepenult stress

ʃo.ko.'la “chocolate”

'pa.ʒi.na “page”

'y.ni.sɛf “Unicef” (unexceptional under standard analysis)

'al.ma.nɛk “almanac” (unexceptional under standard analysis)

As illustrated in (65) above, the QS and the QI hypothesis with respect to the default rule in Dutch stress differ in the division that they make between rule and exceptions. The QI analysis entails that, for words that end in a Light-Heavy sequence, like for any other words, penult stress is the rule, and antepenult stress is exceptional. The QS analysis, on the other hand, entails that for words that end in a Light-Heavy sequence, unlike for other words, antepenult stress is the rule, and penult stress is exceptional.

4.2.1.3 Psycholinguistic evidence

Data from nonce-word testing with adult speakers (Nouveau 1994, Ernestus and Neijt 2008, Domahs et al. 2014) as well as children (Nouveau 1994) provide evidence in favor of the QS pattern described by van der Hulst (1984), Kager (1989), Nouveau (1994), van Oostendorp (1997), and others:

(66) Standard QS analysis of Dutch main stress assignment

1. XLH and XXLH words receive antepenultimate stress
2. All other words receive penultimate stress

Speakers’ judgements are stochastically distributed, but a clear tendency towards more antepenultimate stress in Light-Heavy final items and penultimate stress in other items (especially Heavy-Light final ones) emerges, as will be detailed below. I attribute this tendency to the default grammar (as is done in van der Hulst 1984, Kager 1989, Nouveau

1994, van Oostendorp 1997, 2012, Gussenhoven 2014) – see the beginning of section 4.1 for references to experimental work that suggests that speakers’ judgments are dominated by a default grammar. Nouveau (1994) provides some evidence of this sort for children acquiring Dutch stress. The fact that native speakers’ judgments for stress are stochastic, as in Table 33, Table 34, and Table 35, means that speakers are influenced by the exceptions to some extent when giving their judgments.

It appears that this influence of exceptions on judgments happens on the basis of many factors, including perceived origin of a nonce word. For instance, impressionistically, there seems to be a correlation between exceptional antepenultimate stress and “foreignness”, as evidenced by loanwords like [ˈgrɛ.fə.ti] from English [ɡræ.ˈfɪ.ti] “graffiti”, and [ˌtɛ.sa.ˈlo.ni.ki] (some speakers) from Greek [θɛ.sa.lo.ˈni.ki] “Thessaloniki”; see also footnote 29 below.

Because of these complex factors, I will not attempt to model the stochastic aspect of speakers’ judgments on non-words. Rather, the goal for the learning model will be for it to assign the pattern in (66) above to novel items, as opposed to, for instance, the all-penultimate pattern proposed by van Oostendorp (2012) (see (65) in section 4.2.1.2 above). However, it will be important in future work to connect the default grammar generated by the learner to the stochastic judgment pattern given by learners; see section 4.5 for ideas on how to do this. For the time being, I will simply take the psycholinguistic evidence reviewed here as evidence for the default grammar’s generating the pattern in (66).

Nouveau (1994) presents a body of psycholinguistic evidence on Dutch stress in addition to her theoretical analysis. She probed main stress assignment with a series of

production tasks with both adults and children. Adults were presented with written forms in isolation and asked to read these out loud. Children (3 years old and 4 years old) were asked to repeat words presented to them auditorily in a play context (see Nouveau 1994:121-2 for details). Unfortunately, both experiments were done with a short nonce word list, which limits the degree to which these results can be generalized, but these preliminary results are confirmed by later research done by Ernestus and Neijt (2008) and Domahs et al. (2014).

The experiment with adults reveals a stochastic preference for antepenult stress in Light-Heavy final words, a stochastic preference for penult stress in Heavy-Heavy final words, and an absolute preference for penult stress in Heavy-Light final words. The results are summarized in the table below. Only 3- and 4-syllable words are included in this summary, and the percentages reflect are averaged over the words in the category (the number of items for each category is indicated after the category name; 20 participants gave one judgment for each word).

Table 33. Adult judgments from Nouveau (1994): percentage of antepenultimate, penultimate, and final stress among nonce word pronunciations, per weight type

Weight pattern (number of items)	No. of words			Weight pattern (number of items)	No. of words		
	Antep. stress	Penult stress	Final stress		Antep. stress	Penult stress	Final stress
X X L L (1)	5%	90%	5%	X L L (2) ²⁷	40%	25%	35%
X X L H (1)	30%	40%	30%	X L H (2)	60%	7.5%	32.5%
X X H L (2)	0	100%	0	X H L (0)	-	-	-
X X H H (2)	20%	45%	35%	X H H (0)	-	-	-

Both 3 year old and 4 year old children were presented with words in different stress patterns, and their success rates at imitating these words with these stress patterns were recorded; success means correct imitation of both segmental content and stress²⁸. There was exactly one segmental string for each weight type, which was presented with every possible main stress pattern (including pre-antepenultimate stress).

Table 34 shows, for the word types under consideration here, the percentage of correct imitations across subjects. Note that percentages for each row do not sum to 100%. The first line in each cell reports data for 3 year olds, the second line reports data for 4 year olds.

²⁷ The two words of type X L L differ widely in their stress judgments: /fenimo/ received 75% antepenult stress and 5% penult stress, whereas /faxyri/ received 5% antepenult stress and 45% penult stress. My personal speculation is that this might correlate with the fact that the first word looks like a generic Latinate word, while the second word looks like a French word.

²⁸ Nouveau (1994) reports rates of incorrect imitation, from which these success rates were computed.

Table 34. Success rates of imitating main stress patterns in nonce words per weight pattern for 3 and 4 year old children. Derived from error rate tables (48) in Nouveau (1994:130) and (50) *ibid*:133

Weight pattern (number of items)	No. of words		
	Antep. stress	Penult stress	Final stress
X X L L (1)	70% 70%	80% 95%	35% 25%
X X L H (1)	65% 65%	55% 35%	45% 45%
X X H L (0)	-	-	-
X X H H (0)	-	-	-

Weight pattern (number of items)	No. of words		
	Antep. stress	Penult stress	Final stress
X L L (1)	70% 90%	90% 75%	60% 70%
X L H (1)	80% 95%	60% 55%	85% 90%
X H L (1)	25% 35%	90% 95%	35% 20%
X H H (1)	35% 55%	70% 75%	30% 60%

As can be seen in Table 34, both age groups show a higher rate of imitation of antepenult stress and a lower rate of imitation of penult stress (X)XLH items. XHL and XHH items show a clear preference for penult stress. (X)XLL words mostly show a preference for penult stress – except for XLL words in 4 year old children, which prefer antepenult stress (this is not unlike the pattern found by the learner without a phonotactic learning stage, as will be shown in section 4.4.3.2).

Nouveau also shows that the pattern of errors in the child data points towards regularization. The majority of incorrect imitations, including segmental errors, of non-regular stress patterns (i.e., those that do not follow the rule laid out in section 4.2.1.1) tend towards regularization.

For instance, words in Nouveau's category of relative unmarked exceptions (Light-Light final words with antepenult stress and Heavy final forms with final stress) are corrected to a rule-obeying pattern 64% of the time for 3 year old children, and 58% of the time for 4 year old children. On the other hand, incorrect imitations of regular stress items retain a regular stress pattern 71% of the time for 3 year old children, and

60% of the time for 4 year old children. These numbers are further evidence that the pattern in (66) is on the right track for an abstract lexicon-wide grammatical analysis of Dutch stress.

Other experiments found in the literature corroborate Nouveau's findings. Ernestus and Neijt (2008) present data from a paper-and-pencil experiment in which subjects were presented with nonce words in Dutch orthography, and subjects were asked to mark stress in these words. The nonce words were 3 or 4 syllables long, with all non-final syllables being Light. The final syllable varied between Light and Heavy. A total of 120 different nonce words were constructed, with each subject being given a sample of 40 out of these.

The results, aggregated over subjects and normalized per weight pattern, are shown in Table 35 below. This table shows, once again, a higher tendency to assign antepenult stress and a lower tendency to assign penult stress in Light-Heavy final words.

Generalized linear mixed models with multivariate normal random effects show highly significant results on the comparison between Light-Light final words and Light-Heavy final words: $p < 0.001$ both if subjects are treated as a random effect and if experimental items are treated as a random effect.

Interestingly, Ernestus and Neijt also find a significant difference between 3 and 4-syllable words in the same models ($p < 0.05$ if subjects are a random effect, $p < 0.01$ if experimental items are a random effect). As can be seen in the table, 3-syllable words ending in Light-Heavy have a majority preference for antepenultimate stress, whereas 4-syllable words with the same weight pattern do not – there is simply a not a preference for penultimate stress. Since I am only interested in finding a divide between grammar

and exceptions at a high level of description, I will not attempt to model this difference between 3 and 4-syllable words.

Table 35. Adult judgments Ernestus and Neijt (2008): percentage of antepenultimate, penultimate, and final stress among nonce word pronunciations, per weight type

Weight pattern	No. of words			Weight pattern	No. of words		
	Antep. stress	Penult stress	Final stress		Antep. stress	Penult stress	Final stress
X X L L	36.7%	69.2%	4.2%	X L L	31.4%	62.8%	5.9%
X X L H	44.3%	44.8%	10.9%	X L H	57.2%	32.5%	10.3%
X X H L	-	-	-	X H L	-	-	-
X X H H	-	-	-	X H H	-	-	-

Domahs et al. (2014) present data from a production experiment in which adults read 3-syllable nonce words in a carrier sentence. Nonce words are controlled for similarity to existing words by excluding any potential words whose last two syllables rhyme with existing words attested in CELEX (Baayen, Piepenbrock, and Gulikers 1995).

Unfortunately, no raw results of counts of stress pattern per weight pattern are available. However, Domahs et al. show that both a regression analysis and a hierarchical clustering analysis of the data find that the weight of both the penultimate and the final syllable has a highly significant effect on the choice between antepenultimate and penultimate stress.

Thus, the various pieces of experimental evidence that shed light on the mental representation of Dutch stress agree that there is a structural effect which shifts stress towards the antepenultimate syllable when the word ends in a Light-Heavy sequence, with penultimate stress generally being preferred. For this reason, I will assume that the traditional QS analysis of Dutch stress, as described in section 4.2.1.1, is on the right track, and I will see this analysis as the goal for the simulations described in this chapter.

The fact that Dutch primary stress can be analyzed in terms of either a QI analysis

or a QS analysis, while only a QS analysis is learned by language-acquiring infants, is an interesting case study for exception induction. An adequate model of learning exceptions should project a grammar which has the QS default rule illustrated in (66) from the data given in Table 32 above.

The learners presented in this chapter will be subjected to precisely this test: they will be given data based on those in Table 32 above, and the grammar resulting from learning will be tested on previously unseen words to determine the default rule that results from these grammars. These simulations will be described in section 4.4, where it will be shown that at least the learners with a phonotactic learning stage (Jarosz 2006a) attain the desired result. However, before this, I will lay out an OT analysis of Dutch main stress assignment based Nouveau (1994) in section 4.2.2, in order to create a better understanding of the phonological explanation of this stress pattern.

4.2.2 OT analysis of Dutch main stress

For the simulations of this chapter, I will adopt a version of Nouveau's (1994) analysis of Dutch primary stress. Nouveau's analysis of the default main stress pattern differs minimally from later analyses of Dutch stress (van Oostendorp 1997, Gussenhoven 2009) – these analyses diverge from Nouveau's in other aspects. The version of Nouveau's analysis of the default stress pattern adopted here will be reviewed in section 4.2.2.1. I will indicate any deviations from Nouveau's original analysis.

Nouveau assumes that exceptional stress patterns are obtained by inverting the ranking of certain constraints for certain lexical items. This is very similar to idea in indexed constraint theory (see section 4.1) that an indexed constraint can invert the ranking of two constraints for a set of lexical items. In section 4.2.2.2, I will show a

variant of Nouveau's (1994) proposal for accounting for all types of exceptions to the rule described in sections 4.2.1 and 4.2.2.1.

4.2.2.1 Non-exceptional stress

On a high level, Nouveau's (1994) analysis builds on the idea expressed by van der Hulst (1984) and Kager (1989) that stress falls on the rightmost non-final foot head of a word. Feet are assumed to be trochaic and QS with CVC syllables being heavy.

In a Light-Light final words, as in (67a), the rightmost foot in the word has its head on the penultimate syllable, which is not final, so that it receives main stress. However, since feet are QS, it is not allowed to build a disyllabic foot over the last two syllables of a Light-Heavy final words, as in (67b). Instead, the rightmost foot of the word must be on the last syllable only – but it may not receive main stress, because main stress may not be on the final syllable, as shown in (67c).

Instead, main stress falls on the preceding foot head. Since the penultimate syllable is light, this means that the preceding foot head is the antepenultimate syllable, as in (67d). In this manner, words ending in Light-Heavy receive antepenultimate main stress.

(67) Antepenultimate stress in Light-Heavy final words

- a. pi('ja.ma)
- b. *y('ni.sɛf) heavy syllable must be a foot head
- c. *(,y.ni)('sɛf) [sɛf] is word-final
- d. ('y.ni)(,sɛf)

All other types of words receive penultimate stress. If the final syllable is not heavy, as in (68ab), it is not necessary or possible to make this final syllable a foot head, but instead, a foot is built over the last two syllables, which leads to penultimate stress.

If the penultimate syllable is heavy, it will always be a foot head. As shown in

(68c), this leads to penult main stress, even when the final syllable is heavy.

(68) Penultimate stress in all words that do not end in Light-Heavy

- a. Light-Light final words: pi('ja.ma) *('pi.ja)(,ma)
- b. Heavy-Light final words: a('χɛn.da) *('a.χɛn)(,da)
- c. Heavy-Heavy final words: e('lɛk)(trɔn)²⁹ *('e.lɛk)(,trɔn)

4.2.2.1.1 Implementation

Nouveau (1994) implements the analysis sketched in section 4.2.2.1 above in OT by ranking eight constraints, most of which are taken from the original Prince and Smolensky (1993) manuscript.³⁰ These constraints' definitions are paraphrased in (69) below:

(69) Constraints used in the current analysis (based on Nouveau 1994)

1. Edgemost(R) : One violation mark for every syllable intervening between the right edge of the word and the syllable with main stress.
2. Trochee : One violation mark for every foot whose first syllable is not its head.
3. WSP : One violation mark for every heavy syllable which is not a foot head.
4. Ft-Bin : One violation mark for every foot which does not consist of two syllables or two moras. (Heavy syllables are assigned two moras.)
5. Non-Finality(σ') : One violation mark if the head syllable of the Prosodic Word (i.e., the syllable with main stress) ends the Prosodic Word.
6. Non-Finality(Ĥt) : One violation mark if the head foot of the Prosodic Word (i.e., the foot that contains main stress) ends the Prosodic Word.
7. Parse-syllable : One violation mark for every syllable that is not part of a foot.
8. *Clash : One violation mark for every pair of foot heads that are adjacent to one another.

²⁹ As will be seen in section 4.2.2.1.1, I assume with Nouveau that the last two syllables of a Heavy-Heavy final word are actually footed together in one foot to avoid stress clash: e('lɛk.trɔn). However, this does not matter for the location of main stress.

³⁰ Nouveau also includes the undominated constraint Lx≈PR (lexical words should coincide with prosodic words), as well as an undominated constraint that is only relevant for superheavy syllables, which I will omit here.

If it is assumed that feet are trochaic (i.e., Trochee is undominated), then the fact that penultimate stress is the default can be expressed by ranking Edgemost(R) above Non-Finality($\acute{F}t$) and Parse-syllable. As can be seen in tableau (70), Edgmost(R) is violated more often in the case of antepenultimate stress, while penultimate stress in Light-Light-final words violates Non-Finality($\acute{F}t$) and Parse-syllable. The fact that penultimate stress wins means that Edgemost(R) must be above Non-Finality($\acute{F}t$) as well as Parse-syllable.

(70) Argument for Edgemost(R) >> Non-Finality($\acute{F}t$), Parse-syllable

/pijama/	Edgemost(R)	Non-Finality($\acute{F}t$)	Parse-syll
a. pi('ja.ma)	*	*	*
b. ('pi.ja)(,ma)	**!		

The fact that, in Light-Light final words with exceptional antepenultimate stress, like ['pa.χi.na], the last syllable receives secondary stress³¹ means that Ft-Binarity must be ranked under Parse-syll³². This is because the final secondary stress comes on a light syllable, which, under the assumption that feet are trochaic, means that the presence of this stress entails a monomoraic foot.

(71) Argument for Parse-syll >> Ft-Binarity

/paχina/ (exceptional)	Parse-syll	Ft-Binarity
a. ('pa.χi)na	*!	
b. ('pa.χi)(,na)		*

In order to make sure that Light-Heavy final words receive antepenultimate rather than penultimate stress, Non-Finality($\acute{\sigma}$) and WSP must be ranked above Edgemost(R). This is because final stress, (72a), violates Non-Finality($\acute{\sigma}$), even though it satisfies Edgemost(R), while parsing the final heavy syllable into the same foot as the preceding

³¹ There is some disagreement about judgments of secondary stress in such cases, but I will follow the secondary stress patterns used by Kager (1989) and van Oostendorp's (1997).

³² Nouveau assumes that Ft-Binarity is undominated, but does not present explicit evidence for this. Since *Clash is also a part of the analysis, Ft-Binarity is not needed to ensure alternating stress.

light syllable, as in (72b), violates WSP, even though this has fewer violations of Edgemost(R) than the winning candidate which has antepenultimate stress, (72c).

(72) Argument for Non-Finality($\acute{\sigma}$), WSP >> Edgemost(R)

/ynisɛf/	Non-Finality($\acute{\sigma}$)	WSP	Edgemost(R)
a. (,y.ni)(ˈsɛf)	*!		
b. y(ˈni.sɛf)		*!	*
☞ c. (ˈy.ni)(,sɛf)			**

Furthermore, Non-Finality($\acute{\sigma}$) must be above WSP, in order to derive penult stress in disyllabic words that end in Light-Heavy, such as [ˈro.bɔt]. Constructing a foot around the final heavy syllable, as in (73a), would leave no space for another binary trochaic foot on (ro), which means that there will be final stress. This option violates Non-Finality($\acute{\sigma}$). The attested option, however, which is penult stress, as in (73b), violates WSP. This means that Non-Finality($\acute{\sigma}$) must be ranked above WSP.

(73) Argument for Non-Finality($\acute{\sigma}$) >> WSP

/robɔt/	Non-Finality($\acute{\sigma}$)	WSP
a. ro(ˈbɔt)	*!	
☞ b. (ˈro.bɔt)		*

Since the dataset considered in my simulations will only contain 3 and 4-syllable words, the ranking Non-Finality($\acute{\sigma}$) >> WSP is not essential.

Finally, to capture the fact that in Heavy-Heavy final words, there is penultimate main stress but no secondary stress on the last syllable, *Clash must be ranked above WSP:

(74) Argument for *Clash >> WSP

/rododɛndrɔn/	*Clash	WSP
☞ a. (,ro.do)(ˈdɛn.drɔn)		*
b. (,ro.do)(ˈdɛn)(,drɔn)	*!	

Candidate b. in tableau (74) violates *Clash, because satisfying WSP by giving both heavy syllables in the word their own foot leads to adjacent foot heads. Instead, a

candidate with one single foot over the penult and final syllables wins, which means that *Clash must be ranked over WSP.

These ranking arguments yield the following ranking:

(75) Ranking for non-exceptional stress

Trochee, *Clash, Non-Finality(σ) >> WSP >> Edgemost(R) >> Parse-syllable, Non-Finality(Ĥt)³³ >> Ft-Bin

The tableaux below show how this derives antepenult stress in Light-Heavy final items (that have an antepenultimate syllable), and penult stress in all other cases. Tableau (76) shows how antepenultimate stress is derived, based on the Light-Heavy final word /jeryzalem/ “Jerusalem”. Tableaux (77-79) show how penultimate stress wins in words the end in a Light syllable, or have a penultimate syllable that is Heavy.

(76) Summary tableau for Light-Heavy final /jeryzalem/

/jeryzalem/	Trochee	*Clash	Non-Finality(σ)	WSP	Edgemost(R)	Parse-syll	Non-Finality(Ĥt)	Ft-Bin
a. je(ry.za)(lēm)			*!			*	*	
b. (je.ry)(za.lēm)				*!	*		*	
c. je(ry.za)(lēm)	*!	*			*	*		
d. (jery)(za)(lēm)		*!			*			*
e. je('ry.za)(lēm)					**	*		
f. je('ry.za)lēm				*!	**	**		
g. ('jery)za(lēm)					***!	*		

Tableau (76) shows that final stress, as in candidate a., is ruled out by top-ranked Non-Finality(σ). Penult stress, represented by candidates b.-d., is ruled out either by WSP (if a binary foot is constructed over the last two syllables, as in b.), or by *Clash or Trochee (if the penultimate foot is made iambic to create penultimate main stress, as in c., or if the

³³ Nouveau (1994) also assumes the ranking Parse-syllable >> Non-Finality(Ĥt), but there is no direct evidence for this ranking.

penultimate foot is built only around the penultimate syllable, [ni], as in d.). Pre-antepenultimate stress, however, as in g., is ruled out because of its excessive violation of Edgemost(R). This leaves antepenultimate stress as the only option. The candidate with antepenultimate stress that does not have secondary stress on the last syllable, f., is ruled out because it violates WSP. Therefore, antepenultimate main stress with secondary stress on the last syllable, as in e., surfaces as the winning option.

This situation is different for words that end in a light syllable, such as /makaroni/ “macaroni”:

(77) Summary tableau for Light-Light final /makaroni/

/makaroni/	Trochee	*Clash	Non-Finality(ó)	WSP	Edgemost(R)	Parse-syll	Non-Finality(Ft)	Ft-Bin
a. ma(,ka.ro)('ni)			*!			*	*	*
☞ b. (,ma.ka)('ro.ni)					*		*	
c. ma(ka.'ro)(,ni)	*!	*			*	*		*
d. (,ma.ka)('ro)(,ni)		*!			*			*
e. ma('ka.ro)(,ni)					**!	*		*
f. ma('ka.ro)ni					**!	**		
g. ('ma.ka)(,ro.ni)					**!*			

The difference between tableau (76) and (77) is that placing [ni], a light syllable, in foot-dependent position does not lead to a violation of WSP. Candidates c. and d. are still excluded by top-ranked constraints. Final stress, as in candidate a., is still excluded because of its violation of Non-Finality(ó), but placing main stress on a syllable before the penult, as in candidates e.-g., is penalized by its excessive violations of Edgemost(R).

When the penult and the final syllable are both heavy, as in /rododendrɔn/, penultimate stress is also generated. This is because WSP does not prefer antepenult over

penult stress, or *vice versa*. As can be seen in tableau (78) below, the only candidates that do not have one violation of WSP are c. and d., which have a violation of undominated *Clash. Final stress, as in (78a), is excluded by Non-Finality(σ'), while antepenultimate and pre-antepenultimate stress, as in (78e.-g), are excluded by WSP or Edgemost(R).

(78) Summary tableau for Heavy-Heavy final /rododendron/

/rododendron/	Trochee	*Clash	Non-Finality(σ')	WSP	Edgemost(R)	Parse-syll	Non-Finality(Ŕt)	Ft-Bin
a. ro(,do.dɛn)(,drɔn)			*!	*		*	*	
☞ b. (,ro.do)(,dɛn.drɔn)				*	*		*	
c. ro(do.,dɛn)(,drɔn)	*!	*			*	*		
d. (,ro.do)(,dɛn)(,drɔn)		*!			*			
e. ro(,do.dɛn)(,drɔn)				*	**!	*		
f. ro(,do.dɛn)drɔn				**!	**	**		
g. (,ro.do)(,dɛn.drɔn)				*	**!*			

Finally, in words that end in Heavy-Light, like /propaxanda/, WSP and the ranking Non-Finality(σ') >> Edgemost(R) converge in their preference of penultimate stress. As can be seen in tableau (79), only penultimate (b.-d.) and pre-antepenultimate stress (g.) satisfy WSP. Candidates a. and c.-e. are ruled out because they violate top-ranked constraints, while antepenultimate stress candidate f. is ruled out because of its violation of WSP. Finally, pre-antepenultimate stress, as in g., is ruled out by its excessive violation of Edgemost(R), leaving b. as the winner.

(79) Summary tableau for Heavy-Light final /propaxanda/

/propaxanda/	Trochee	*Clash	Non-Finality(ó)	WSP	Edgemost(R)	Parse-syll	Non-Finality(Ét)	Ft-Bin
a. pro(,pa.xan)(,da)			*!	*		*	*	*
☞ b. (,pro.pa)(,xan.da)					*		*	
c. pro(pa,'xan)(,da)	*!	*			*	*		*
d. (,pro.pa)(,xan)(,da)		*!			*			**
e. pro('pa.xan)(,da)				*	**	*		*
f. pro('pa.xan)da				*!	**	**		
g. ('pro.pa)(,xan.da)					**!*			

Now that I have laid out a version of Nouveau's (1994) treatment of default main stress assignment in Dutch, I will continue with a treatment of exceptional stress. I will closely follow her account, which uses re-ranking constraints in order to account for exceptionality, but I will present it in a manner consistent with Lexically Indexed Constraint Theory.

4.2.2.2 Exceptional stress

As illustrated in Table 32, all weight types (-LL, -LH, -HL, -HH) exhibit antepenultimate, penultimate, and final stress. At the same time, the default pattern specifies that Light-Heavy final words have antepenultimate stress, while other words have penultimate stress. This means that there are three exceptional patterns that must be accounted for:

(80) Exceptional patterns for Dutch stress

1. Final stress
examples: [,kro.ko.'dɪl] (Light-Heavy), [,fo.ko.'la] (Light-Light)
2. Penultimate stress in words that end in Light-Heavy
example: [se.'le.bəs] (Light-Heavy)

3. Antepenultimate stress in words that do not end in Light-Heavy
example: ['pa.χi.na] (Light-Light)

Nouveau derives these exceptional stress patterns by re-ranking certain parts of the hierarchy in (75). Specifically, the following re-rankings are necessary (I have left some of Nouveau's originally proposed re-rankings out of this schema, for reasons that will be discussed below):

(81) Re-rankings for an analysis of exceptional Dutch stress

1. Final stress:
Edgemost(R) >> Non-Finality(σ') instead of
Non-Finality(σ') >> Edgemost(R)
2. Penultimate stress in words that end in Light-Heavy:
Non-Finality(σ') >> Edgemost(R) >> WSP instead of
Non-Finality(σ') >> WSP >> Edgemost(R)
3. Antepenultimate stress in words that do not end in Light-Heavy:
Non-Finality(Ĥt) >> (WSP >>) Edgemost(R) instead of
(WSP >>) Edgemost(R) >> Non-Finality(Ĥt)

Each of these exceptional rankings will be discussed in a separate subsection, with an analysis in terms of indexed constraints.

4.2.2.2.1 Final stress

Final stress, which is exceptional for any weight type in Dutch, can be represented by raising Edgemost(R) above Non-Finality(σ'), even though it is below those constraints and WSP for non-exceptional words. The fact that a constraint must be raised, not lowered, with respect to the unmarked ranking makes it possible to represent this in the framework of indexed constraints: the indexed constraint Edgemost(R)_i is ranked above Non-Finality(σ'), while the non-indexed constraint Edgemost(R) is ranked below WSP.

This produces final stress in words like [ˌfo.ko.'la], as illustrated in tableau (82) below. Candidates c. and d. are excluded because of their violations of Clash and/or

Trochee. For the rest, all candidates with violations of Edgemost(R), which are b., e., and f., are excluded, leaving final stress, shown in a., as the only option.

(82) Final stress for /fokola_i/, indexed to Edgemost(R)_i

/fokola _i /	Trochee	*Clash	Edgemost(R) _i	Non-Finality(σ)	WSP	Edgemost(R)	Parse-syll	Non-Finality(Ĥt)	Ft-Bin
☞ a. (fo.ko)(la)				*				*	*
b. fo('ko.la)			*!			*	*	*	
c. (fo.'ko)(la)	*!	*	*			*			*
d. fo('ko)(la)		*!	*			*	*		**
e. ('fo.ko)(la)			**!			**			
f. ('fo.ko)la			**!			**	*		

This analysis in terms of indexed constraints requires that versions of Edgemost(R) be raised above Non-Finality(σ). However, Nouveau (1994) argues that the ranking Edgemost(R) >> (Ft-Bin,) Non-Finality(σ) is achieved instead by lowering Non-Finality(σ) under WSP and Edgemost(R):

(83) Nouveau's proposal for representing final stress

/fokola/	Trochee	*Clash	WSP	Edgemost(R)	Non-Finality(σ)	Parse-syll	Non-Finality(Ĥt)	Ft-Bin
☞ a. (fo.ko)(la)					*		*	*
b. fo('ko.la)				*!		*	*	
c. (fo.'ko)(la)	*!	*		*				*
d. fo('ko)(la)		*!		*		*		**
e. ('fo.ko)(la)				**!				
f. ('fo.ko)la				**!		*		*

She argues that this is necessary because of the highly marked status of Heavy-Light words with final main stress (Nouveau 1994:174,200-201). Such words are rare in the

lexicon (see Table 32 in section 4.2.1.2), and final (or antepenultimate) stress is rarely given as a judgment for nonce words that end in Heavy-Light (see section 4.2.1.3).

For this reason Nouveau (1994:200-201) claims that Rightmost(R) must remain under WSP at all times, so that even under the exceptional final stress ranking, Heavy-Light final words come out with penultimate stress, as in (84) below. Both final (candidate a.) and antepenultimate (candidates e.-f.) stress is ruled out by WSP. Out of the remaining penultimate stress candidates, the one without violations of Trochee and *Clash, which is b., is chosen as the winner.

(84) Nouveau (1994): high ranking of WSP rules out final stress in Heavy-Light final words

/tabanda/	Trochee	*Clash	WSP	Edgemost(R)	Non-Finality(ó)	Parse-syll	Non-Finality(Ĥt)	Ft-Bin
a. (, ta.ban)(, da)			*!		*		*	*
b. ta('ban.da)				*		*	*	
c. (ta.'ban)(, da)	*!	*		*				*
d. ta('ban)(, da)		*!		*		*		**
e. ('ta.ban)(, da)			*!	**				
f. ('ta.ban)da			*!	**		*		

However, words that end in Heavy-Light with final stress (such as [,fri.kan.'do] “fricandeau”) are actually attested, and their stress patterns must be able to be derived by the grammar somehow. Nouveau appears to assume that there is some mechanism that accounts for such more ‘extreme’ exceptions by overriding the grammar. Such a mechanism could be provided by underlying stress marks and stress Faithfulness; however, see Gussenhoven (2014) for a summary of truly impossible exceptions to Dutch stress: a Faithfulness-based account must be sure to exclude such categorically

impossible stress patterns. Because Heavy-Light final words with penultimate stress are not a true gap, I assume that an indexed version of Edgemost(R) must outrank WSP at least for these words, and the dispreference for final stress in Heavy-Light final words comes from some other source.

4.2.2.2.2 Exceptional penultimate stress

In order to account for penultimate stress in words that end in the sequence Light-Heavy, such as [se.'le.bəs], Nouveau proposes another re-ranking: Edgemost(R) is raised above WSP, but below Non-Finality(σ̇). This simply means that the preference to include the final Heavy syllable in its own foot is less important than having stress more to the right, while final stress is still disallowed.

The result of this ranking is illustrated in the tableau in (85) below: candidate a., which would have won under the ranking for final stress, is now excluded by its violation of Non-Finality(σ̇). Candidates c. and d. are still excluded by top-ranking constraints. However, antepenultimate stress, as in candidates e. and f., is penalized by Edgemost(R), so that candidate b., with penultimate stress, wins despite its violation of WSP.

(85) Nouveau's proposal for exceptional penultimate stress

/selebes/	Trochee	*Clash	Non-Finality(σ̇)	Edgemost(R)	WSP	Parse-syll	Non-Finality(Ŕt)	Ft-Bin
a. (,se.le)(.bəs)			*!				*	
☞ b. se('le.bəs)				*	*	*	*	
c. (se.'le)(.bəs)	*!	*		*				
d. se('le)(.bəs)		*!		*		*		*
e. ('se.le)(.bəs)				**!				
f. ('se.le)bəs				**!	*			

In the framework of Lexically Indexed Constraint Theory, this re-ranking can be

implemented in two possible ways. The first possible analysis posits a second indexed version of Edgemost(R) that is above WSP but below Non-Finality($\acute{\sigma}$). This analysis is shown in tableau (86).

The second possible analysis is that an indexed version of Non-Finality($\acute{\sigma}$) is ranked above indexed Edgemost(R) (which is above WSP). This analysis is worked out in tableau (87).

Tableau (86) shows the option of having two separate indexed versions of the constraint Edgemost(R). One is ranked over WSP but below Non-Finality($\acute{\sigma}$), and one is ranked over Non-Finality($\acute{\sigma}$).

Because of the assumption of nested indexation laid out in section 4.1.1 (every pair of constraints with the same name must be in a stringency relation), the former, because it is lower, must now be called Edgemost(R)_i, while the latter, because it is higher, must now be a recursively indexed constraint Edgemost(R)_{i,j}. Words that end in Light-Heavy that have penultimate stress are indexed to Edgemost(R)_i, but not to Edgemost(R)_{i,j}. Words with final stress are indexed to both (because of the assumption of nested indexation).

The fact that Edgemost(R)_i is ranked below Non-Finality($\acute{\sigma}$) means that final stress, as in (86a), is excluded by Non-Finality($\acute{\sigma}$). At the same time, antepenultimate stress, as in (86e-f), is excluded by Edgemost(R)_i (\neq Edgemost(R)_{i,j}), because it is ranked above WSP. This leaves penultimate stress as the remaining option, with (86b) being the only option that does not violate Trochee or *Clash. Final stress, under this hypothesis, would be generated by indexing final stress words to Edgemost(R)_{i,j}.

(86) Option 1: penultimate stress for /selebəs_i/, indexed to Edgemost(R)_i, whose rank is lower than in (82)

/selebəs _i /	Trochee	*Clash	Edgemost(R) _{i,j}	Non-Finality(ó)	Edgemost(R) _i	WSP	Edgemost(R)	Parse-syll	Non-Finality(Ft)	Ft-Bin
a. (,se.le)('bəs)				*!					*	
☞ b. se('le.bəs)					*	*	*	*	*	
c. (se.'le)(,bəs)	*!	*			*		*			
d. se('le)(,bəs)		*!			*		*	*		*
e. ('se.le)(,bəs)					*!*		**			
f. ('se.le)bəs					*!*	*	**	*		

The other option is to rank an indexed version of Non-Finality(ó), Non-Finality(ó)_k, above indexed Edgemost(R)_i, which is shown in tableau (87). In this case, Light-Heavy final words with penultimate stress must be indexed to both Edgemost(R)_i and Non-Finality(ó)_k, in order to make sure that Non-Finality(ó) outranks Edgemost(R), and Edgemost(R) outranks WSP.

In this tableau, once again, final stress, as in (87a), is ruled out by Non-Finality(ó)_j, as it is ranked above Edgemost(R)_i. The latter constraint, in turn, rules out antepenultimate stress, as in (87e-f), even though the only remain option, penultimate stress as in (87b), has a violation of WSP (candidates c. and d. are ruled out by top-ranking constraints).

(87) Option 2: Penultimate stress for /selebəs_{ij}/, indexed to Edgemost(R)_i and Non-Finality(σ)_j

/selebəs _{ij} /	Trochee	*Clash	Non-Finality(σ) _j	Edgemost(R) _i	Non-Finality(σ)	WSP	Edgemost(R)	Parse-syll	Non-Finality(Ft)	Ft-Bin
a. (se.le)(bəs)			*!		*			*	*	
b. se('le.bəs)				*		*	*		*	
c. (se.'le)(bəs)	*!	*		*			*	*		
d. se('le)(bəs)		*!		*			*			*
e. ('se.le)(bəs)				*!*			**	*		
f. ('se.le)bəs				*!*		*	**	**		

In this scenario, Edgemost(R)_i must be dominated by Non-Finality(σ)_j because, as can be seen in tableau (87), Edgemost(R)_i prefers final stress. If Edgemost(R)_i were not dominated by Non-Finality(σ)_j, then penultimate stress, as in (87b), would not win.

In the remainder of the exposition of the analysis, I will display this second option of accounting, but the first option is equally viable as an analysis of this phenomenon.

4.2.2.2.3 Exceptional antepenultimate stress

Finally, Nouveau argues that exceptional antepenultimate stress can be accounted for by raising Non-Finality(Ft) over Edgemost(R). This is shown for the word ['pa.χi.na] in tableau (88), where Non-Finality(Ft) has an indexed version, Non-Finality(Ft)_k. Candidates a., c., and d. are excluded by top-ranked constraints, and candidate b. with penultimate stress is excluded by its violation of Non-Finality(Ft). The only remaining candidates are e. and f., which have antepenultimate main stress. Candidate f. is excluded because of its violation of Parse-syllable, so that candidate e., with secondary stress on

the last syllable, wins³⁴.

(88) Antepenultimate stress for /paχina_k/, indexed to Non-Finality(Ĥt)_k

/paχina _k /	Trochee	*Clash	Non-Finality(ó)	WSP	Non-Finality(Ĥt) _k	Edgemost(R)	Non-Finality(Ĥt)	Parse-syll	Ft-Bin
a. (,pa.χi)(,na)			*!		*		*		*!
b. pa(,χi.na)					*!	*	*	*	
c. (pa.,χi)(,na)	*!	*				*			*
d. pa(,χi)(,na)		*!				*		*	*
e. (,pa.χi)(,na)						**			*
f. (,pa.χi)na						**		*!	

Nouveau claims that there is a separate exceptional ranking Parse-syll >> WSP that is needed to derive antepenultimate stress in Heavy-Heavy final words. However, as I show in tableau (89) below for the nonce word [,ta.lak.,tan] used in Nouveau's experiment, the fact that *Clash is top-ranked makes it so that antepenultimate is also derived for this type of words if they are indexed to Non-Finality(Ĥt)_k. This tableau also includes the other indexed constraints used in this analysis.

³⁴ Since Nouveau assumes an undominated ranking for Ft-Binarity, candidate f. wins under her analysis.

(89) Antepenultimate stress for Heavy-Heavy final /talaktan_k/, indexed to Non-Finality($\acute{F}t$)_k

/talaktan _k /	Trochee	*Clash	Non-Finality($\acute{\sigma}$) _j	Edgemost(R) _i	Non-Finality($\acute{\sigma}$)	WSP	Non-Finality($\acute{F}t$) _k	Edgemost(R)	Non-Finality($\acute{F}t$)	Parse-syll	Ft-Bin
a. (, ta.lak)(, tan)					*!	*	*		*		
b. ta('lak.tan)						*	*!	*	*	*	
c. (ta.'lak)(, tan)	*!	*						*			
d. ta('lak)(, tan)		*!						*		*	
e. ('ta.lak)(, tan)						*		**			
f. ('ta.lak)tan						**!		**		*	

Finally, the fact that some Heavy-Light final words occur with antepenultimate stress can be accounted for by introducing the additional ranking Non-Finality($\acute{F}t$)_m >> WSP, as is shown in tableau (90) below. As in tableau (89), candidates c. and d. are ruled out by top-ranked constraints, while candidate a. is ruled out by Non-Finality($\acute{\sigma}$). However, the choice between penultimate stress (candidate b.) and antepenultimate stress (candidates e. and f.) is made by the mutual ranking of Non-Finality($\acute{F}t$)_k and WSP. The former constraint prefers antepenultimate stress, while the latter prefers penultimate stress. Since antepenultimate stress is the desired outcome, Non-Finality($\acute{F}t$)_k must be above WSP.

(90) Antepenultimate stress in Heavy-Light final words: Non-Finality($\acute{F}t$)_k >> WSP

/talakta _k /	Trochee	*Clash	Non-Finality($\acute{\sigma}$) _j	Edgemost(R) _i	Non-Finality($\acute{\sigma}$)	Non-Finality($\acute{F}t$) _k	WSP	Edgemost(R)	Non-Finality($\acute{F}t$)	Parse-syll	Ft-Bin
a. (ta.lak)(ta)					*!	*	*		*		*
b. ta('lak.ta)						*!		*	*	*	
c. (ta.'lak)(ta)	*!	*						*			*
d. ta('lak)(ta)		*!						*		*	*
e. ('ta.lak)(ta)							*	**			*
f. ('ta.lak)ta							*	**		*!	

4.2.2.2.4 Summary of analysis

In conclusion, the three types of exceptional stress in Dutch (given the default rule described in section 4.2.1.1) can be accounted for with three indexed constraints. Final stress, as in [_Jo.ko.'la], can be accounted for by indexing the relevant words to Edgemost(R)_i, which is to be ranked above Non-Finality($\acute{\sigma}$), as shown in section 4.2.2.2.1. Exceptional penultimate stress in Light-Heavy final words, as in [se.'le.bes], can be accounted for by indexing these words to Non-Finality($\acute{\sigma}$)_j, which is to be ranked above Edgemost(R)_i, as shown in section 4.2.2.2.2.³⁵

Finally, exceptional antepenultimate stress in words that do not end in Light-Heavy, as in ['pa.χi.na], can be accounted for by indexing the relevant words to Non-Finality($\acute{F}t$)_k, which is to be ranked above Edgemost(R) (and WSP), as shown in section 4.2.2.2.3. (91) below briefly schematizes the resulting analysis:

³⁵ One alternative account would be having two versions of Edgemost(R) with the ranking Edgemost(R)_{i,j} >> Non-Finality($\acute{\sigma}$) >> Edgemost(R)_i >> WSP. In this case, words with exceptional antepenultimate stress are indexed to Edgemost(R)_i, while words with exceptional final stress are indexed to both Edgemost(R)_i and Edgemost(R)_{i,j}.

(91) Summary of lexically indexed constraint analysis of Dutch stress

a. Ranking

Trochee, *Clash, **Non-Finality**($\acute{\sigma}$)_j >>

Edgemost(R)_i >>

Non-Finality($\acute{\sigma}$), **Non-Finality**($\acute{F}t$)_k >>

WSP >>

Edgemost(R) >>

Non-Finality($\acute{F}t$), Parse-syllable >>

Ft-Binarity

b. Words with final stress indexed to Edgemost(R)_i

c. Words with exceptional penultimate stress indexed to Non-Finality($\acute{\sigma}$)_j

d. Words with exceptional antepenultimate stress indexed to Non-Finality($\acute{F}t$)_k

At first glance, this analysis appears to be overly permissive: antepenultimate, penultimate, and final stress are all possible for any combination of syllable weights. However, there are two ways in which this analysis is restrictive. First, only one stress pattern is possible for every weight pattern in words that are not marked with at least one index (*i*, *j*, or *k*).

Second, pre-antepenultimate stress remains impossible (see Kager 1989 and Gussenhoven 2014)³⁶. As can be seen in tableaux below, indexing a four-syllable word to either Non-Finality($\acute{\sigma}$)_j or Non-Finality($\acute{F}t$)_k cannot yield pre-antepenultimate stress (Edgemost(R)_i can only prefer stress shift to the right, and will therefore not be examined).

As is shown in tableau (92), pre-antepenultimate stress, as in candidate g., loses even when a word is indexed to Non-Finality($\acute{\sigma}$)_j, because it has the more violations of Edgemost(R) than the winning candidate – candidate b. – but an equal amount of violations of higher ranked constraints (in this case, a single violation of WSP).

³⁶ Unfortunately, the other categorically impossible stress patterns mentioned by Gussenhoven (2014) involve segmental interactions with stress that fall outside the scope of the current investigation.

Penultimate stress wins because final stress, as in a., is excluded by Non-Finality($\acute{\sigma}$)_j, while antepenultimate or pre-antepenultimate stress (candidates e. and g.) are excluded by Edgemost(R). Candidate f. is harmonically bounded by candidate e., and candidates c. and d. are excluded by top-ranked Trochee and/or *Clash.

(92) Failed attempt at generating pre-antepenultimate stress 1: index /malabandan_j / to Non-Finality($\acute{\sigma}$)_i

/malabandan _j /	Trochee	*Clash	Non-Finality($\acute{\sigma}$) _j	Edgemost(R) _i	Non-Finality($\acute{\sigma}$)	Non-Finality($\acute{F}t$) _k	WSP	Edgemost(R)	Parse-syll	Non-Finality($\acute{F}t$)	Ft-Bin
a. ma(,la.ban)('dan)			*!		*		*		*	*	
☞ b. (,ma.la)('ban.dan)							*	*		*	
c. ma(la. 'ban)(,dan)	*!	*						*	*		
d. (,ma.la)('ban)(,dan)		*!						*			
e. ma('la.ban)(,dan)							*	**!	*		
f. ma('la.ban)dan							**!	**	**		
g. ('ma.la)(,ban.dan)							*	**!*			

As shown in tableau (93), pre-antepenultimate stress loses for the same reason when a four-syllable word is indexed to Non-Finality($\acute{F}t$)_k. Candidates c. and d. violate Trochee and/or *Clash, candidate a. is excluded by Non-Finality($\acute{\sigma}$), while candidate b., which won in tableau (92), is now excluded by Non-Finality($\acute{F}t$)_k. Pre-antepenultimate stress, as in candidate g., loses from antepenultimate stress, as in candidate e., because of its excessive violation of Edgemost(R). Candidate f. is harmonically bounded by e..

(93) Failed attempt at generating pre-antepenultimate stress 1: index /malabandan_k / to Non-Finality($\acute{F}t$)_k

/malabandan _k /	Trochee	*Clash	Non-Finality($\acute{\sigma}$) _j	Edgemost(R) _i	Non-Finality($\acute{\sigma}$)	Non-Finality($\acute{F}t$) _k	WSP	Edgemost(R)	Parse-syll	Non-Finality($\acute{F}t$)	Ft-Bin
a. ma(, la.ban)('dan)					*!	*	*		*	*	
b. (, ma.la)('ban. dan)						*!	*	*		*	
c. ma(la. 'ban)(, dan)	*!	*						*	*		
d. (, ma.la)('ban)(, dan)		*!						*			
e. ma('la.ban)(, dan)							*	**	*		
f. ma('la.ban)dan							**!	**	**		
g. ('ma.la)(, ban.dan)							*	***!			

Thus, the analysis summarized in (91) covers all attested stress patterns in monomorphemic words of 3 and 4 syllables, while the grammar covers exactly the pattern in (66). Moreover, the impossible exceptional stress pattern of pre-antepenultimate stress (Kager 1989, Gussenhoven 2014) is excluded by the analysis.

4.3 The learning framework: inducing lexically indexed constraints

The simulations in this chapter are set up in the framework of Expectation Driven Learning (EDL; Jarosz submitted). This framework applies Expectation Maximization (EM; Dempster, Laird, and Rubin 1977) to the learning of ranked constraint grammars, and has been successful as an account of hidden structure learning problems (Jarosz 2006a, submitted).

EDL, whose mechanics will be detailed in section 4.3.1, relies on pairwise comparisons between constraints to arrive at a ranking, as opposed to both the Constraint Demotion type of approaches (Tesar 1995) and error-driven approaches such as the Gradual Learning Algorithm (Boersma 1998) and Stochastic Gradient Ascent (Soderstrom, Mathis, and Smolensky 2006).

This reliance on pairwise rankings makes it easy to pin down the exceptionality of a word to a certain constraint pair – which is much more difficult in the approaches mentioned above. Pater (2010) shows that the Recursive Constraint Demotion approach to inducing indexed constraints (see also Becker 2007 and Coetzee 2009b) is also liable to run into such problems in the case when inconsistency happens between more than two constraints at once. The approach to pinning down exceptionality to a constraint and a group of lexical items proposed here will be laid out in section 4.3.2.

As mentioned in section 4.1.1, constraint indexation is assumed to be recursive. Every word that is found to be exceptional with respect to a particular constraint is initially indexed to one and the same version of that constraint. For instance, if the lexicon prefers $A \gg B$, every word that prefers $B \gg A$ is indexed to the same indexed constraint, B_i . If more a fine-grained division of these exceptions into different classes is necessary, a recursively indexed constraint B_{ij} is induced for the exceptional class among those exceptions, as was also illustrated in section 4.1.1.

The learner consists of three modules: a generator module and an Expectation Maximization module, which are taken from the original EDL model and will be described in section 4.3.1, and an exception induction module, which will be described in section 4.3.2. The learner also makes use of a phonotactic learning stage (Jarosz 2006a), which I will address briefly in section 4.3.3. Finally, I will briefly summarize the algorithm in section 4.3.4.

4.3.1 Expectation Driven Learning

For the simulations reported on in this chapter, I extended upon the batch version of the EDL framework as described in Jarosz (submitted). I will briefly describe this

framework in this subsection, before moving on to the model of indexed constraint induction that was used.

4.3.1.1 Pairwise ranking probabilities

The EDL framework defines constraint ranking in terms of probabilities over pairwise rankings, as illustrated in Table 36, Table 37, and Table 38 below. During learning, these probabilities are fed into a sampling procedure to generate fully ordered rankings of the relevant constraints.

The probabilities associated with a constraint pair influence how often that constraint pair will be ranked one way or the other. For instance, if $p(A \gg B) = 1$, constraint A will always be above B (unless $B \gg A$ is entailed by other choices made by the sampling procedure; see Jarosz submitted for details). However, if $p(A \gg B) = p(B \gg A) = 0.5$, A will be above B roughly 50% of the time. Finally, if $p(A \gg B) = 0.7$, then A will be above B roughly 70% of the time, while B will be above A in all remaining cases. For any constraint pair $\{C, D\}$, the probabilities of $C \gg D$ and $D \gg C$ always sum to 1.

Other approaches to stochastically variable ranking have also been proposed – see Anttila (1997), Boersma (1998), and Coetzee (2009a). However, none of these approaches directly assess pairwise rankings.

The fact that each pairwise ranking has a certain probability of being set one or the other way, which can be seen as a stochastic generalization of Partially Ordered Grammars (Anttila 2002), allows the model to make maximal use of stochastic tendencies found when processing the data.

Examples of pairwise ranking probabilities are provided below. Table 36

represents the categorical ranking $A \gg C \gg B$. Table 37 represents the variable ranking $\{A, C\} \gg B$, where A has a 50% chance of being above C , but both constraints are categorically above B . Finally, Table 38 represents a situation in which $A \gg C$ and $C \gg A$ are equally likely, but C is more likely to be above B than A is: $C \gg B$ has a probability of 80% and $A \gg B$ has a probability of 70%.

Table 36. Example of probabilistic grammar for EDL: Categorical grammar $A \gg C \gg B$

Ranking	Probability	Ranking	Probability
$A \gg B$	1	$B \gg A$	0
$A \gg C$	1	$C \gg A$	0
$B \gg C$	0	$C \gg B$	1

Table 37. Example of probabilistic grammar for EDL: Categorical ranking $A, C \gg B$; ranking of A and C probabilistic

Ranking	Probability	Ranking	Probability
$A \gg B$	1	$B \gg A$	0
$A \gg C$	0.5	$C \gg A$	0.5
$B \gg C$	0	$C \gg B$	1

Table 38. Example of probabilistic grammar for EDL: All rankings are probabilistic

Ranking	Probability	Ranking	Probability
$A \gg B$	0.7	$B \gg A$	0.3
$A \gg C$	0.5	$C \gg A$	0.5
$B \gg C$	0.2	$C \gg B$	0.8

4.3.1.2 The generator

The learner's generator module takes a probability distribution over pairwise rankings like the ones illustrated in Table 36, Table 37, and Table 38 above, and samples a categorical ranking from it. This sampling is done by choosing a random order of all numeric cells in the distribution table, after which the algorithm goes through the cells one by one, choosing 1 or 0 according to the probability in the cell. For instance, if a cell has a probability of 0.8, it will be replaced by 1 with a probability of 80%, and 0 with a probability of 20%.

In order to avoid generating logically inconsistent rankings (for instance, $A \gg B$,

$B \gg C$, $C \gg A$), the generator checks at every step whether the pairwise ranking just decided on entails any other rankings, and enforces these entailments in the ranking table. For instance, if the generator has set $A \gg B$ and $B \gg C$ to 1, then $A \gg C$ is set to 1 and $C \gg A$ is set to 0. These transitivity entailments can be computed in polynomial time: see Jarosz (submitted) for more details on how these computations proceed.

Once a categorical ranking has been computed, it is used to choose a winning candidate in the tableau that is it offered. This winning candidate is then compared to the attested winner in the training data, and a match or mismatch is assessed. In this manner, the generator produces a match or mismatch for a particular training datum given a set of pairwise ranking probabilities.

4.3.1.3 Expectation Maximization

The Expectation Maximization module in the learner estimates the probabilities of pairwise rankings in the grammar by computing them from the previous grammar and the corpus of data, as formulated in (94). See Jarosz (submitted) for more details on this part of the learner, and for its grounding in the literature on Expectation Maximization outside of phonology and/or linguistics.

(94) Updating constraint pair probabilities through Expectation Maximization
For every constraint pair $\{A, B\}$:

$$p(A \gg B | g_{t+1}) = p(A \gg B | D, g_t)$$

After being initialized with a grammar in which all pairwise ranking probabilities are 0.5 (although other initializations are also possible), the learner loops through the Expectation Maximization update rule in (94) until a certain number of iterations is reached, or until a criterion of matching the training data is satisfied.

The probability $p(A \gg B | D, g_t)$, referenced in the formula in (94), is calculated

by a Bayesian computation, shown in (100), for which the necessary ingredients consist of the probability of $A \gg B$ in the grammar, $p(A \gg B | g_t)$, as well as the probability of the data given $A \gg B$ within the current grammar and the probabilities of the data given $B \gg A$ within the current grammar. In order to be able to obtain the latter two probabilities, two variants are made of the current grammar g_t : one variant in which the probability of $A \gg B$ is replaced by 1, and one variant in which the probability of $B \gg A$ is replaced by 1.

Multiple samples are taken from both grammars, and the learner counts how many times either ranking produces the correct outcome for a given data point. Since multiple samples are taken from the grammar, this probes the space of probability defined by the grammar.

If the counts are divided by the size r of the sample taken, this allows for estimating the probability of a given data point d given the current grammar g with the ranking $A \gg B$ fixed: $\hat{p}(d | A \gg B, g)$, and its counterpart – the probability of data point d given the current grammar g with the ranking $B \gg A$ fixed: $\hat{p}(d | B \gg A, g)$:

(95) Computation of the conditional probability of a data point given a fix pairwise ranking and a grammar

Given a data point d in the data set D , a pair of constraints $\{A, B\}$, a grammar g , and a sample size r :

Let $\text{match}(d, m) = 1$ iff the sample ranking m sampled from g produces a candidate that has the correct surface form for d ; else $\text{match}(d, m) = 0$.

$$\hat{p}(d | A \gg B, g) = \frac{\sum_{i=1}^r \text{match}(d, m_i)}{r} \text{ where } m \text{ is drawn from } g \text{ with } p(A \gg B) = 1$$

$$\hat{p}(d | B \gg A, g) = \frac{\sum_{i=1}^r \text{match}(d, m_i)}{r} \text{ where } m \text{ is drawn from } g \text{ with } p(B \gg A) = 1$$

In practice, to shorten learning time, a shortcut was used, which is not a part of the

original EDL algorithm as presented by Jarosz (submitted). When a constraint has the same number of violations for every candidate, its ranking with respect to other constraints cannot be determined, since it does not prefer any candidate over any other candidate. For instance, the constraint WSP (see (69) in section 4.2.2.1) has zero violations for all candidates if the input has no Heavy syllables. This is also true for any indexed constraint and a word that is not marked for that constraint: for instance, Max_i in example (57) in section 4.1 has 0 violations for any candidate of the input /tat/, which is not marked for index i .

For this reason, when a constraint pair contains a constraint that has the same number of violations for the input currently under consideration, the learner simply skips that constraint pair for that input. This is represented in the model by setting the probability of generating that input under either ranking of that constraint pair to 0.

(96) Shortcut for computing $\hat{p}(d|A \gg B, g)$

For any constraint pair $\{A, B\}$ and input word d :

If A or B has the same number of violations for all of d 's candidates:

$$\hat{p}(d|A \gg B, g) = \hat{p}(d|B \gg A, g) = 0$$

Using Bayes' Law, the probabilities of pairwise rankings given a data point and a grammar, computed as shown above, can be transformed to produce the probability of each pairwise ranking given the data point and the current grammar: $p(A \gg B|d, g)$, $p(B \gg A|d, g)$, as shown in (97).

(97) Computation of the conditional probability of a pairwise ranking given data point d

$$p(A \gg B|d, g) = \frac{\hat{p}(d|A \gg B, g) * p(A \gg B|g)}{p(d|g)} \text{ (Bayes' Law)}$$

The probability $p(A \gg B|g)$ can be found directly in the grammar g . The probability of the data point given the grammar can be computed by marginalization:

(98) Computing $p(d|g)$

$$p(d|g) = \hat{p}(d|A \gg B, g) * p(A \gg B|g) + \hat{p}(d|B \gg A, g) * p(B \gg A|g)$$

If, for any reason, $p(d|g)$ equals zero (which, for example, would be the case when the shortcut in (96) is used), the value of $p(A \gg B|d, g)$ is set to 0.5. This includes words for which A or B has an equal number of violations for every candidate. The logic behind this is that, whenever $p(d|g)$ is greater than zero, $p(A \gg B|d, g) = 0.5$ if $\hat{p}(d|A \gg B, g) * p(A \gg B|g) = \hat{p}(d|B \gg A, g) * p(B \gg A|g)$:

(99) Situation: $p(\text{data}|\text{ranking}) * p(\text{ranking})$ yields same nonzero value for both rankings
If $\hat{p}(d|A \gg B, g) * p(A \gg B|g) = \hat{p}(d|B \gg A, g) * p(B \gg A|g) = 0.8$, then

$$p(A \gg B|d, g) = \frac{0.8}{p(d|g)} = \frac{0.8}{0.8 + 0.8} = 0.5$$

The version of EDL used for the current simulations is a batch version, so that the probability of $A \gg B$ is only updated after all data have been seen, as shown in the formula in (94). The probability of $A \gg B$ given all data and the current grammar, expressed as $p(A \gg B|D, g)$, is computed very similarly to the analogous probability for a single word, $p(A \gg B|d, g)$:

(100) Probability of a pairwise ranking given the entire data set D

$$p(A \gg B|D, g) = \frac{\sum_{d \in D} p(d|A \gg B, g) * p(A \gg B|g)}{p(D|g)}$$

The probability of the entire data corpus given the current grammar, $p(D|g)$, is computed by summing $p(d|g)$ for all data points:

(101) Computing $p(D|g)$

$$\begin{aligned} p(D|g) &= \sum_{d \in D} p(d|g) \\ &= \sum_{d \in D} p(d|A \gg B, g) * p(A \gg B|g) + p(d|B \gg A, g) * p(B \gg A|g) \end{aligned}$$

Since words for which every candidate has an equal number of violations for A and B have $p(d|A \gg B, g) = p(d|B \gg A, g) = 0$, the sum in (100) is not updated at all for such constraint pair and word combinations. This is especially helpful if one of the constraints in the pair is an indexed constraint: if the current word is not indexed to the constraint, and thus has zero violations for that constraints in all its candidates, the word is ignored for the purpose of updating the ranking probabilities of the indexed constraint with respect to other constraints.

In order to update every constraint pair's ranking probability, the learner computes $p(d|A \gg B, g)$ and $p(d|B \gg A, g)$ for every word d in the data set and for every constraint pair $\{A, B\}$. These values are then used to compute $p(d|A \gg B, g)$ and $p(d|B \gg A, g)$ for every constraint pair $\{A, B\}$ as indicated in (100) and (101).

Finally, as indicated in (94) above, the grammar's pairwise ranking probabilities are updated by making them equal to the probabilities of these same rankings given the entire dataset and the previous grammar.

The pseudocode below summarizes the entire Expectation Maximization module:

(102) Pseudo-code summary of the batch EDL algorithm (Jarosz submitted)

Initialize ranking probabilities

For all constraint pairs $\{A, B\}$, $p(A \gg B|g) = p(B \gg A|g) = 0.5$

Repeat until learning criterion is reached:

1. Estimate pairwise ranking probabilities

For all data point d in data set D and all constraint pairs $\{A, B\}$:

If A or B has the same number of violations for all of d 's output candidates:

$$p(d|A \gg B, g) = p(d|B \gg A, g) = 0$$

Else:

Let $g(A \gg B)$ be the current grammar g with $p(A \gg B|g)$ set to 1

Take r samples from $g(A \gg B)$

Invoke **generator module** – count the number of successes

Compute $p(d|A \gg B, g)$: $\frac{\text{number of successes}}{r}$

Let $g(B \gg A)$ be the current grammar g with $p(B \gg A|g)$ set to 1

Take r samples from $g(B \gg A)$

Invoke **generator module** – count the number of successes
 Compute $p(d|B \gg A, g)$: $\frac{\text{number of successes}}{r}$

2. Update grammar
 For all constraint pairs $\{A, B\}$:
 Compute $p(A \gg B|D, g)$ through Bayes' Law:
 For all words d in D , sum $p(d|A \gg B, g)$
 Multiply the result by $p(A \gg B|g)$
 Divide by $p(D|g)$ (see (101))
 Update $p(A \gg B|g_{\text{new}}) = p(A \gg B|g_{\text{old}})$
 Update $p(B \gg A|g_{\text{new}}) = 1 - p(A \gg B|g_{\text{old}})$

Now that the Expectation Maximization module of EDL has been explained, I will turn to the extension of EDL proposed in this chapter: the exception induction module, which induces and updates indexed constraints.

4.3.2 Inducing indexed constraints

In this subsection, I present my own addition to EDL: an exception induction module. This module is based on ideas advanced by Pater (2010), in particular the idea that indexed constraints are induced whenever a data point prefers a ranking opposite to that preferred by the entire data set. However, Pater (2010) defines this opposition of rankings categorically, in terms of inconsistency in the context of Recursive Constraint Demotion (Tesar 1995).

Inconsistency is the situation where one word in the data can only be explained given a certain ranking (e.g., $A \gg B$), while another word in the data can only be explained given a contradictory ranking (e.g., $B \gg A$). Recursive Constraint Demotion is set up in a way to automatically detect such situations, and once inconsistency is detected, the learning procedure stops.

As Pater (2010) points out, the induction of lexically indexed constraints could apply in exactly this situation: inconsistency detection signals that there is a contradiction between words as to how a certain pair or group of constraints is ranked, while induction

of indexed constraints can solve inconsistency by assigning the words that need the contradictory ranking to an indexed constraint. For instance, if certain words need $A \gg B$ while others need $B \gg A$, then the latter group of words can be indexed to an indexed version of B , and the ranking $B_i \gg A \gg B$ will allow to sidestep the contradiction.

However, as also pointed out by Pater (2010), inconsistency detection only detects the presence of a contradiction, but does not localize that contradiction to a set of words and set of pairwise rankings. Pater suggests some ways to localize inconsistency to a pair of constraints and a set of words, but a procedure that is guaranteed to find which constraint to induce for which word proves to be elusive.

The learner presented here responds to these challenges by taking a stochastic route, and specifically one within the framework of EDL. The exception induction module finds exceptions by comparing lexicon-wide tendencies for a given constraint pair to word-specific tendencies for that constraint pair.

As shown in section 4.4.1.3 above, it is possible to compute the probability of a pairwise ranking given a single data point as well as given the entire dataset, as shown in (97) and (100), respectively. This makes it possible to estimate whether a given word prefers a ranking of a given pair of constraints that is opposite to the lexicon-wide preference, and induce lexically indexed constraints based on these opposite preferences.

The exception induction module is invoked at every iteration of the Expectation Maximization module, unless the learner is currently in a phonotactic learning stage (see section 4.3.3). Since the exception induction module needs access to the probability of rankings given the entire data set, it needs to apply after the entire lexicon is examined, but before the ranking probabilities in the grammar are updated (cf. the pseudocode in

(102)).

The structure of the exception induction module at a high level is as follows. For each word and for each constraint pair, it gauges whether the word's preference for the ranking of this constraint pair, represented by $p(A \gg B|d, g)$, computed as in (97), is opposite to the lexicon-wide preference, represented by $p(A \gg B|D, g)$, computed as in (100). Then it induces or updates an indexed constraint for the constraint pair that has the strongest overall difference in ranking preferences between the lexicon-wide trend and exceptions.

Since rankings in the current model are stochastic, there must be some criterion to decide whether the ranking preferences of a word and of the entire lexicon differ due to random error or due to underlying exceptionality. To maximally simplify the model, a cutoff point was used for this purpose. Given a threshold parameter α , the following definition was used:

(103) Criterion for opposite pairwise ranking preferences

Word d and data set D ($d \in D$) have opposite preferences for constraint pair $\{A, B\}$ iff

$$p(A \gg B|D) \geq 0.5 + \alpha$$

and

$$p(B \gg A|d) \geq 0.5 + \alpha$$

For the simulations described here, α was chosen to be 0.1. This means that if a word prefers $B \gg A$ with a probability of at least 60% (i.e., $p(A \gg B|d) \leq 0.4$), while the entire dataset prefers $A \gg B$ with a probability of at least 60% (i.e., $p(A \gg B|D) \geq 0.6$), the word is seen as having a preference opposite from the lexicon as a whole.

If it was found that a word has a ranking preference opposite to that of the entire

lexicon with respect to a constraint pair, this word was marked an exception for the purpose of that constraint pair. For instance, if the lexicon as a whole prefers Non-Finality($\acute{\sigma}$) \gg Edgemost(R) with at least 60%, while the word [,sa.ban.'da] prefers Edgemost(R) \gg Non-Finality($\acute{\sigma}$) with at least 60%, the word [,sa.ban.'da] is marked as an exception for the constraint pair {Edgemost(R), Non-Finality($\acute{\sigma}$)}.

It is the constraint that is preferred to be lower by the lexicon that was cloned for the purposes of indexed constraint induction, because of the stringency condition discussed in section 4.1.1. For instance, if the lexicon prefers Non-Finality($\acute{\sigma}$) \gg Edgemost(R) while [,sa.ban.'da] prefers Edgemost(R) \gg Non-Finality($\acute{\sigma}$), the constraint induced will be Edgemost(R)_i, with index *i* assigned to /sabanda/. This constraint can derive the correct stress pattern for [,sa.ban.'da] if it is ranked above Non-Finality($\acute{\sigma}$): Edgemost(R)_i \gg Non-Finality($\acute{\sigma}$) \gg Edgemost(R).

The criterion for actually inducing a constraint was as follows. For all words that are marked exceptional for constraint pair {A,B} – I will designate those words $X_{\{A,B\}}$ –, the absolute difference between $p(A \gg B|D)$ and $p(A \gg B|d)$ was summed. The most exceptional constraint pair is the one with the highest summed $|p(A \gg B|D) - p(A \gg B|d)|$ over all exceptions. This is summarized in (104) below.

(104) Indexed constraint induction

At every iteration, for every constraint pair {A,B}, and all words $X_{\{A,B\}} \subset D$ found to be deviant for that constraint pair:

If $\sum_{d \in X_{\{A,B\}}} |p(A \gg B|D) - p(B \gg A|d)|$ is greater than for any other constraint pair:³⁷

If $p(A \gg B|D) > 0.5$, induce B_i and index it to the words in $X_{\{A,B\}}$

If $p(A \gg B|D) < 0.5$, induce A_i and index it to the words in $X_{\{A,B\}}$

When an indexed constraint is chosen to be induced, but such a constraint already exists,

³⁷ If there is more than one maximally exceptional constraint pair, one of them is chosen at random for indexed constraint induction.

then the pre-existing indexed constraint is simply updated with whichever words were not previously indexed to it. For instance, if the procedure calls for inducing an indexed constraint $\text{Edgemost}(R)_i$ that applies to a group of words including /sabanda/, but there is already a pre-existing indexed constraint $\text{Edgemost}(R)_i$ that does not apply to /sabanda/, then /sabanda/ is simply added to the group of words indexed to the pre-existing $\text{Edgemost}(R)_i$.

If there is no pre-existing indexed constraint of the required kind exists, the constraint is added to the tableaux for every word in the data (with zero violations for words not indexed to these constraints). For instance, if $\text{Edgemost}(R)_i$ is to be induced and there is no such constraint in the constraint set, it is simply added to the constraint set, with regular violations for the words that it is indexed to, and zero violations for all other words.

The place in the probabilistic hierarchy given to the new constraint, B_i , is the exact same as that of the constraint that it is derived from, B . For instance, $p(B_i \gg A) = p(B \gg A)$. The mutual ranking of B and B_i is left in the middle: $p(B_i \gg B) = 0.5$. This is done by way of zero hypothesis: the most conservative guess as to the ranking of B_i is that it has the same ranking as the constraint that it is derived from. Any evidence for differences in ranking between B and B_i should emerge from the Expectation Maximization module.

Because the burden of disentangling the mutual ranking of B and B_i is put entirely on the learner, and the learner updates constraint ranking probabilities gradually, it may take several iterations for B_i to have a high enough ranking to accounting for $\{A, B\}$'s exceptional words. It is for this reason that only the maximally exceptional constraint pair

is chosen for indexed constraint induction or updating at every iteration, as in (104).

After B_i has gained a high enough ranking, words that prefer $B \gg A$ will no longer strongly disprefer the ranking $A \gg B$ – their preference of $A \gg B$ will start approximating 0.5. For this reason, $\{A, B\}$ will no longer be the maximally exceptional constraint pair. However, as long as $\{A, B\}$ is still the maximally exceptional constraint, there is no guarantee that B_i has taken effect.

For this reason, the criterion in (104) makes sure that one indexed constraint is absorbed by the grammar before inducing another one. This creates a bias against inducing several indexed constraints for the same goal (although, as will be seen in the results in section 4.4.3.4, this bias does not prevent that situation: several indexed constraints for the same class of exception are still induced).

The entire indexed constraint updating and induction procedure is formulated in pseudocode in (105):

(105) Pseudocode summary of indexed constraint extension to EDL

Induce B_i and index it to the words in X :

If B_i indexed to the words in Y already exists in the constraint set:

Let pre-existing B_i be indexed to $X \cup Y$

Else:

Add B_i to the constraint set;

For all constraints $C \neq B$, set $p(B_i \gg C) = p(B \gg C)$;

Set $p(B_i \gg B) = p(B \gg B_i) = 0.5$

Since indexed constraints are part of the regular constraint set, recursive indexation, as introduced in section 4.1.1, is an option. Some subset of the words indexed to B_i can turn out to exhibit exceptional behavior, and this subset can be indexed to another index, j . The recursively constraint B_{ij} is then created – from which another recursively indexed constraint B_{ijk} can also be created, if necessary. As mentioned in section 4.1.1, this might be necessary if some subpart of the words that require the ranking $B \gg A$ also require

the ranking $B \gg E$, while the other words that require the ranking $B \gg A$ require the ranking $E \gg B$.

To summarize, the exception induction module gauges whether words are potentially exceptional for a constraint pair (by estimating whether word d has at least a 60% preference for $B \gg A$ while the data set D as a whole has at least a 60% preference for $A \gg B$). The constraint pair which has the most net exceptionality (measured by the summed absolute difference between $p(A \gg B|D)$ and $p(A \gg B|d)$ for all words d that are outliers) is then used to induce or update an indexed constraint.

For the constraint pair with the most exceptionality, its deviant words are indexed to whichever member of the constraint pair is preferred to be lower by the entire lexicon. For instance, if the lexicon prefers $A \gg B$, then the words which prefer $B \gg A$ are indexed to B_i with the hope that B_i may later be ranked above A to yield $B_i \gg A \gg B$. If B_i does not yet exist in the grammar, then it is induced, otherwise it is merely updated with the deviant inputs for that constraint pair.

4.3.3 Phonotactic learning

Following Jarosz (2006a), the learner was given the possibility of a period of phonotactic learning, which means that, during a certain period at the outset of learning, any kind of contrast between lexical items is ignored, and only a lexicon-wide grammar is learned. As pointed out by Hayes (2004) and Prince and Tesar (2004), introducing contrast between underlying forms (in the form of Faithfulness) into the grammar at the outset of learning can lead to a failure to learn phonotactic patterns.

Lexically indexed constraints have the same function as Faithfulness in the model pursued here: they encode differences between words that cannot be reduced to their

surface shape. However, it is also possible to fail to learn lexicon-wide patterns by explaining every phonological difference between words in terms of indexed constraints. For instance, all XLH words in the Dutch grammar could be indexed to a constraint that prefers antepenultimate stress, so that the lexicon-wide grammar is free to ignore that quantity-sensitive aspect of Dutch stress (see sections 4.2.1.1 and 4.2.1.3).

This makes the presence of a phonotactic learning stage a potentially useful tool to prevent the attribution of patterns that hold across the lexicon to a particular subset of lexical items, and ensure that the appropriate grammar is learned for a case like Dutch stress. At the same time, as pointed out by Hayes (2004), Jarosz (2006a), and work cited there, there is evidence that language-acquiring infants acquire phonotactic knowledge before they start differentiating between words. This makes the phonotactic learning stage an *a priori* likely component of the learning path.

A phonotactic learning stage was implemented in the current learner by denying the learner the possibility of inducing or updating indexed constraints for a certain number of iterations. Since the data for the simulations contain no Faithfulness constraints, this means that the grammar will have no access to differences between individual words for the duration of the phonotactic learning stage.

In the simulations presented in section 4.4, a phonotactic learning stage of either 0 or 40 iterations (out of a maximum of 80 iterations) is chosen. This means that the learner either never refrains from inducing indexed constraints if the conditions for it are met, or refrains from it during the first 40 iterations of the simulation. In this way, the effects of the phonotactic stage can be examined independently of the properties of the exception induction model.

4.3.4 Summary of learner

To summarize, the algorithm proposed here is built on Expectation Driven Learning, an Expectation Maximization-based learner for ranked constraint grammars proposed by Jarosz (submitted). The grammars used in this learner consist of probabilities over pairwise rankings of constraints, as explained in 3.1.1. EDL itself consists of a generator module (see section 4.3.1.2) and an Expectation Maximization module (see section 4.3.1.3). The latter module iterates over all data points in the data set, and, given an initial set of ranking probabilities, sets every constraint pair's ranking probabilities in accordance to each ranking's likelihood given the data set and the previous grammar (see (94) in section 4.3.1.3).

An exception induction module, proposed in this chapter, is invoked before updating the grammar's ranking probabilities (the reasons for this are detailed in section 4.3.2). The exception induction module induces or updates indexed constraints based on constraint pairs that show conflict between ranking preferences of the entire dataset versus those of individual words. The constraint pair which shows the largest amount of such conflict (see (104) in section 4.3.2 for a definition of this) leads to the creating or updating of an indexed version of the constraint that the data set as a whole prefers to be lower. For instance, if the data set as a whole prefers $A \gg B$, then an indexed version of constraint B is either added to the constraint set, or the set of words indexed to it is updated with the words that are exceptional with respect to $A \gg B$.

A phonotactic learning stage (Hayes 2004, Prince and Tesar 2004, Jarosz 2006a) can help the learning of the regular stress pattern (which is a phonotactic pattern). During this phonotactic learning stage, the learner essentially assumes that any pattern it finds is

applicable to the entire lexicon (see Jarosz 2006a for a more precise formulation in terms of Expectation Maximization). As was detailed in section 4.3.3, this means that the learner does not induce indexed constraints until a certain point in learning has been reached.

To distinguish the effects of this phonotactic learning stage from the effects of indexed constraints learning, the learning was given two options: no phonotactic learning stage, or phonotactic learning for half of the maximum number of iterations.

(106) Pseudocode summary of the entire algorithm

Initialize ranking probabilities:

For all constraint pairs $\{A,B\}$, $p(A \gg B) = p(B \gg A) = 0.5$

Repeat until learning criterion is reached:

0. Phonotactic stage is either always off, or on iff 40 iterations have not yet passed

1. Estimate pairwise ranking probabilities

For all data point d in data set D and all constraint pairs $\{A,B\}$:

Let $g(A \gg B)$ be the current grammar g with $p(A \gg B)$ set to 1

Take r samples from $g(A \gg B)$

Invoke **generator module** – count the number of successes

Compute $p(d|A \gg B, g): \frac{\text{number of successes}}{r}$

Let $g(B \gg A)$ be the current grammar g with $p(B \gg A)$ set to 1

Take r samples from $g(B \gg A)$

Invoke **generator module** – count the number of successes

Compute $p(d|B \gg A, g): \frac{\text{number of successes}}{r}$

2. Invoke **exception induction module** (see 3.2)

If phonotactic learning stage is over:

a. Find which words are outliers for which constraint pair:

word d is exceptional for constraint pair $\{A,B\}$ iff

$p(A \gg B|d) \text{ and } p(B \gg A|D) \geq 0.5 + \text{threshold value}$

b. Find the constraint pair with the highest divergence from the lexicon-wide tendency:

Find constraint pair $\{F,G\}$ which maximizes

sum over outliers: $|p(F \gg G|\text{outlier}) - p(F \gg G|D)|$

c. Index the outliers for the constraint pair to a version whichever one of $\{F,G\}$ the lexicon prefers to be lower

If $p(F \gg G) > 0.5$, index to F

If F_i does not exist yet:

- Induce F_i and index $\{F, G\}$'s outliers to it
 - If F_i does already exist:
 - Make sure that all of $\{F, G\}$'s outliers are indexed to F_i
 - If $p(G \gg F) > 0.5$, index to G
 - (Same procedure as above but with G_i)
- 3. Update grammar
 - For all constraint pairs $\{A, B\}$:
 - Compute $p(A \gg B | D, g)$:
 - Update $p(A \gg B | g_{\text{new}}) = p(A \gg B | D, g_{\text{old}})$
 - Update $p(B \gg A | g_{\text{new}}) = 1 - p(A \gg B | g_{\text{new}})$
- 4. Add new indexed constraints to grammar
 - For all newly induced indexed constraints B_i :
 - For all constraints $C \neq B$, set $p(B_i \gg C) = p(B \gg C)$;
 - Set $p(B_i \gg B) = p(B \gg B_i) = 0.5$

This algorithm will be used to learn the Dutch data summarized in section 4.2 above.

Section 4.4 below will explain the details of the simulations and present results.

4.4 Simulations and results

4.4.1 Training data

The training data offered to the learner are a processed version of the corpus of stress patterns that occur in monomorphemic 3 and 4-syllable words in Dutch. The frequencies of stress patterns per weight pattern shown in Table 32 in section 4.2.1.2 were scaled down by a factor of 10, and rounded to integers. The resulting frequencies are shown in Table 39. Note that this transformation means that the learner is never exposed to words that end in two Heavy syllables.

Table 39. Frequencies of words in learning data per weight and stress type (based on real Dutch frequencies in Table 32)

Weight pattern	No. of words		
	Antep. stress	Penult stress	Final stress
X X L L	2	2	0
X X L H	0	1	1
X X H L	0	1	0
X X H H	0	0	0

Weight pattern	No. of words		
	Antep. stress	Penult stress	Final stress
X L L	6	7	2
X L H	5	1	3
X H L	0	4	1
X H H	0	0	0

This scaling down was done because each occurrence of a stress pattern and a weight pattern was considered as a separate word with a separate input. This was done so that the exception induction module had the opportunity to succeed by indexing all relevant exceptional inputs to an indexed constraints, but could also potentially fail by not indexing every exceptional input to an indexed constraint.

To keep the size of the learning data manageable and to make sure that the result of exception induction is readable by the naked eye, it is better to have a number of inputs under 50, rather than 379 separate inputs to consider (which is the total number of words counted in Table 32). However, there is no reason why the learner could not be applied to the full set of monomorphemic 3 and 4-syllable words in Dutch shown in Table 32. Since the learner does the same number of comparisons and computations for each individual input word, as can be verified in (106), learning time is linear in the number of input words.

Since the particular segmental content (other than consonant versus vowel) of the words does not matter to the generalizations sketched for Dutch, I constructed fictional words for each of the cells in Table 39, to stand in for actual Dutch words. Thus, as per Table 39, there were 6 words of the type X L L with antepenult stress, as in (107a), 4 words of the type X H L with penult stress, as in (107b), and so on. All words had light

syllables in antepenultimate and pre-antepenultimate position. A list of these words is provided in Appendix C.

(107) Examples of words in the learning data

a. X L L with antepenult stress:

['la.ba.,da]

['ma.ba.,da]

['pa.ba.,da]

etc.

b. X H L with penult stress:

[pa.'ban.da]

[ta.'ban.da]

[ka.'ban.da]

etc.

Data were given with strictly alternating secondary stress (i.e., antepenultimate stress means secondary stress on the final syllable, penultimate stress means secondary stress on the pre-antepenultimate syllable, if one is present; Kager 1989).

In the tableau for each word, every possible main stress pattern was considered, including pre-antepenultimate stress for 4-syllable words. For instance, for the word /malabanda/, the main stress variants ['ma.la.ban.da], [ma.'la.ban.da], [ma.la.'ban.da], and [ma.la.ban.'da] were considered.

The main and secondary stress patterns in the candidates were the product of various metrical parses of the same word. The parses considered were all ways to divide a words into feet such that only feet at the edge of a word can be unary. In the extended version, these were the following footings:

(108) Footings in extended version of the data

a. Four-syllable words

(ma.la) (ban.da)

(ma) (la.ban) (da)

(ma) (la.ban) da

ma (la.ban) (da)

ma (la.ban) da

b. Three-syllable words

(la) (ban.da)

la (ban.da)

(la.ban) (da)

(la.ban) da

Within these parses, all possible ways to assign main and secondary stress were considered, which yields a total of 20 parses for 4-syllable words, and 12 parses for 3-syllable words. As shown in Appendix D, this yields 5 parses for each main stress location in 4-syllable words, and 4 parses for each main stress location in 3-syllable words.

Finally, the constraint set used for the simulations is the same as the one used by Nouveau (1994), as described in section 4.2.2, and with the definitions given in (69):

(109) Constraints used for learning Dutch stress with EDL

1. Edgemost(R)
2. Trochee
3. WSP
4. Ft-Bin
5. Non-Finality($\acute{\sigma}$)
6. Non-Finality(\acute{Ft})
7. *Clash
8. Parse-syllable

4.4.2 Simulation setup

The training data described above were learned with the learner described in section 4.3. Simulations were run with sample size = 50 and exception threshold = 0.1, with up to 80 iterations per run. Two types of simulation were run for both data set described above (the extended and the compact data set): simulations were run without a phonotactic learning stage, and with a phonotactic learning stage of 40 iterations. Each type of simulation was run 5 times.

Simulations were stopped whenever the maximum number of iterations had been reached, or when the fit of the grammar to the training data was 95% or more. The fit of the grammar to the training data was computed by sampling 20 rankings from the current grammar, finding outputs for each training word, and computing the proportion of hits

(outputs that match a learning datum) out of the total number of outputs generated.

In order to test the grammar that was learned, an additional set of data was constructed. The purpose of these test data was to see how the grammar performs on items that are not indexed to one of the indexed constraints in the grammar – in other words, how it would perform with novel items. The test data consisted of one item for each possible weight type (XXLL, XXLH, XXHL, XXHH, XLL, XLH, XHL, XHH), with candidates constructed according to the exact same procedure as for the training data (see section 4.4.1 above).

To assess the grammar's performance on these test data, another 20 rankings were sampled from the grammar, and the number of times that each candidate in each test tableau emerged as the winner was recorded. A summary of the outcome of this test for the simulations performed will be provided in the results section below.

4.4.3 Simulation results

The results of these simulations are summarized here. Three aspects are important: the accuracy of the learner on the training data, the way in which the resulting grammars generalize to novel items, and the indexed constraints found by the learner.

As I will show in this section, the simulations with a phonotactic learning stage perform very well in all of these categories: they are almost 90% accurate on the training data on average, they perform well on unseen data, and they are successful at finding indexed constraints for just the exceptional forms. At the same time, while the simulations without a phonotactic learning stage have an equal or slightly better accuracy on the training data and are also successful at finding indexed constraints for just the forms their analyses see as exceptional, they predict the wrong stress pattern in

previously unseen 3-syllable X L L and X H H words. I will show that this is mainly due to finding an erroneous ranking of Parse-syllable and Ft-Binarity, which most probably arises from insufficient comparison between 3 and 4-syllable words before indexed constraints are found.

In section 4.4.3.1, I will discuss how accurate the grammars learned are on the training data. Then, in section 4.4.3.2, I will show the grammars resulting from learning perform on the novel items described in section 4.4.2 above. In section 4.4.3.3, I will evaluate the phonotactic rankings that underlie this wug-performance.

Then, in section 4.4.3.4, I will discuss how the resulting grammars perform on the training data, and which indexed constraint analyses are chosen for this. In section 4.4.3.4.1, I will present a detailed analysis of a high-performance result, where both accuracy on the training data and accuracy on the test data are high. Then, in section 4.4.3.4.2, I will present a similar analysis of a ‘word-case scenario’ result, where accuracy on both training data and test data is low within the set of results, and show that even this result is reasonable. Finally, in section 4.4.3.5, I will briefly summarize the findings obtained in this case study.

4.4.3.1 Accuracy on training data

The simulations without a phonotactic learning stage exhibited an average accuracy of 92.0% on the training data, whereas the simulations with a phonotactic learning stage had an average accuracy of 88.8% on the training data. Thus, both types of simulations performed about equally well on the training data, with a slight advantage for the simulations without a phonotactic learning stage.

4.4.3.2 Testing on novel items

The tables below show results from 20 sample rankings out of the final grammar for both types of simulation: without a phonotactic learning stage, and with a phonotactic learning stage. The times out of these 20 rankings that each main stress pattern emerges as the winner are represented as a percentage in these tables. Cells that represent the desired pattern described in section 4.2.1 are shaded.

The results shown here are averaged over 2 runs of each simulation for each type of simulation. For each type of simulation, both runs had similar results.

Table 40. Results from testing grammars obtained without phonotactic learning

Weight pattern	No. of words		
	Antep. stress	Penult stress	Final stress
X X L L	0%	100%	0%
X X L H	90%	10%	0%
X X H L	0%	100%	0%
X X H H	0%	100%	0%

Weight pattern	No. of words		
	Antep. stress	Penult stress	Final stress
X L L	97%	3%	0%
X L H	100%	0%	0%
X H L	12%	88%	0%
X H H	89%	11%	0%

Table 41. Results from testing grammars obtained with 40 iterations of phonotactic learning

Weight pattern	No. of words		
	Antep. stress	Penult stress	Final stress
X X L L	2%	98%	0%
X X L H	96%	4%	0%
X X H L	0%	98% ³⁸	0%
X X H H	2%	98%	0%

Weight pattern	No. of words		
	Antep. stress	Penult stress	Final stress
X L L	13%	87%	0%
X L H	99%	1%	0%
X H L	7%	93%	0%
X H H	58%	42%	0%

As can be seen in Table 40, the grammars obtained without phonotactic learning produce antepenultimate stress in Light-Heavy words with an average probability of 95%, while they produce penultimate stress in all other word types with an average probability of

³⁸ Two of the five runs produced pre-antepenultimate stress for an X X H L novel word on one of 20 samples from the grammar.

67%, with a total accuracy of 74% on the expected stress pattern (shaded cells).

In fact, it can be seen that these grammars produce a rather different pattern in 3-syllable words. Instead of a preference for penult stress in XLL and XHH words, there is a 93% preference for antepenult stress in these word types. I will show in section 4.4.3.2 that this stems from too high a position of Parse-syllable in the hierarchy: penultimate stress in 3-syllable words is impossible to achieve without either a stress clash – (,pa)(‘ba.da) – or an unparsed syllable – pa(‘ba.da).

On the other hand, the grammar obtained with a phonotactic learning stage produces the desired pattern, as shown in Table 41: Light-Heavy final words have an average preference of 97.5% to have antepenultimate stress, while all other words have an average preference of 86% to have penultimate stress, with a total accuracy of 88.9% on the expected stress pattern (shaded cells).

Three-syllable words ending in two Heavy syllables seem to be an outlier for these grammars: they only receive penultimate stress in just over 40% of all cases. I will show in section 4.4.3.2 below that this, once again, arises from a slightly higher position for Parse-syllable than necessary, and I suggest that this stems from the lack of XHH data in the dataset used here (which, in turn, originates in their low type frequency in the lexicon).

4.4.3.3 Evaluating rankings

I will present here sample rankings for the results of each type of simulation. Although the results do not specify a single ranking, but, rather, a distribution over rankings, a typical ranking can be approximated by ordering constraints by their total probability of dominating some other constraint. This probability is computed by

summing, for a given constraint C, all the probabilities of C over some other constraint: $p(C \gg A) + p(C \gg B) + p(C \gg D) + \dots$. This sum is divided by the total number of constraints that it could dominate (which is the total size of the constraint set Φ minus 1), as shown in (110). Ordering constraints by this probability $p(C \gg \dots)$ gives a sense of the relative height of the constraint in the hierarchy.

(110) Computing domination probability $p(C \gg \dots)$

For a constraint C, element of the total constraint set Φ :

$$p(C \gg \dots) = \frac{\sum_{D \in \Phi, D \neq C} p(C \gg D)}{|\Phi| - 1}$$

Given two sample ranking probability sets, one for each type of simulation, which will be shown in full in Table 44 and Table 45, these yield the following “typical rankings” when constraints are ordered by their $p(C \gg \dots)$ ³⁹:

Table 42. Typical ranking for the first run of the simulation without a phonotactic learning stage

Constraint	*Clash	WSP	Trochee	Non-Fin(σ)	Parse-syll	Ft-Bin	Edge most(R)	Non-Fin(Ét)
$p(C \gg \dots)$	0.99	0.76	0.73	0.57	0.5	0.29	0.16	0

Table 43. Typical ranking for the first run of the simulation with a phonotactic learning stage

Constraint	*Clash	WSP	Non-Fin(σ)	Trochee	Ft-Bin	Edge most(R)	Parse-syll	Non-Fin(Ét)
$p(C \gg \dots)$	0.95	0.79	0.73	0.55	0.54	0.22	0.22	0.00

However, these “typical rankings” do not determine which constraints are ranked categorically with respect to one another. For this purpose, I provide the Hasse diagrams in Figure 11 and Figure 12 below. In these diagrams, rankings with a probability of at least 0.95 is represented with a solid line, and rankings with a probability of at least 0.80 are represented with a dotted line.

³⁹ A dotted line was drawn between two constraints if their $p(C \gg \dots)$ values differ by less than 0.05.

Figure 11. Hasse diagram for the first run of the simulation without a phonotactic learning stage

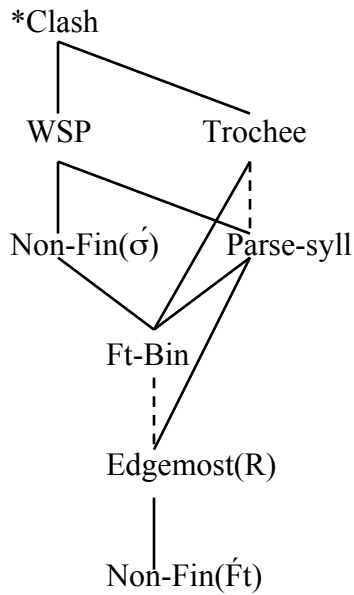
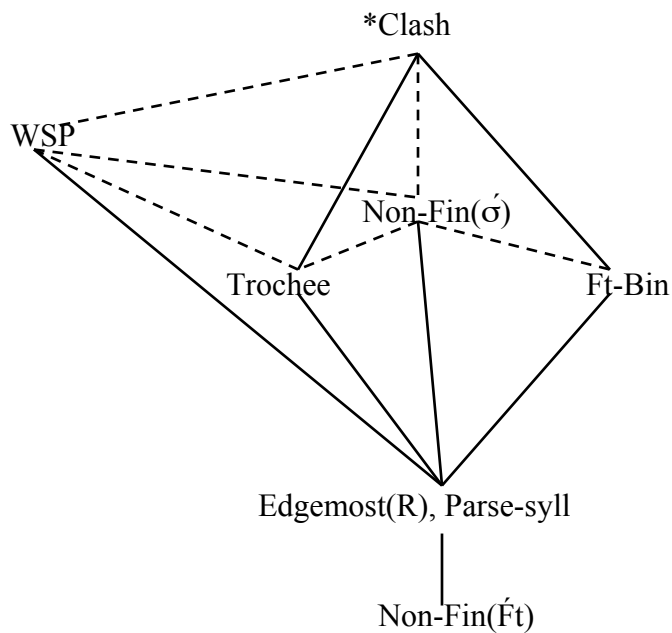


Figure 12. Hasse diagram for the first run of the simulation with a phonotactic learning stage



As can be seen in these diagrams, the two types of simulation yield similar results: *Clash and WSP are high-ranked, Non-Fin(σ) and Ft-Bin are somewhere in between, Parse-syll is dominated by WSP and Trochee, and Edgemost(R) and Non-Fin(Ĥt) are at

the bottom, with Edgemost >> Non-Fin($\acute{F}t$).

However, there are several noticeable differences. One of these differences lies in the mutual ranking of Ft-Bin, Parse-syll, and Edgemost(R).

The simulation without a phonotactic learning stage ranks Parse-syll above Ft-Bin and Edgemost(R). I will show in section 4.4.3.3.1 below that the ranking Parse-syll >> Ft-Bin, Edgemost(R) leads to the erroneous assumption of antepenultimate stress in XLL and XHH words in the outcome of this type of simulation. I will argue that the ranking Parse-syll >> Ft-Bin is acquired because exceptions are induced before the trends that the 3 and 4-syllable words have in common are sufficiently processed by the learner.

On the other hand, the simulation with a phonotactic learning stage ranks Parse-syll below Ft-Bin, which is desirable (cf. section 4.2.2.1), but Edgemost(R) and Parse-syll can be ranked either way (the probability of Edgemost(R) >> Parse-syll is about 0.5). I will show in section 4.4.3.3.2 that, when Parse-syll is above Edgemost(R), antepenultimate stress is generated. Furthermore, I will argue that Edgemost(R) >> Parse-syll is not learned because of the absence of XHH forms in the data presented to the learner.

Finally, I will show the complete rankings tables for both learners in section 4.4.3.3.3.

4.4.3.3.1 Erroneous antepenultimate stress in learner without phonotactic stage

The ranking Parse-syll >> Ft-Bin, Edgemost(R), observed for the simulation with a phonotactic learning stage, leads to erroneous antepenultimate main stress in XLL and XHH words, as illustrated in the tableaux in (111-112) below.

Candidates c. and d. in both tableaux are excluded because they violate top-

ranked *Clash. Of the remaining candidates, final stress (candidate a.) is excluded because of its violation of Non-Fin(σ'), while penultimate stress (candidate b.) is excluded because of its violation of Parse-syll. Antepenultimate stress without secondary stress on the last syllable is ruled out either by Parse-syll (tableau (111)), or by WSP (tableau (112)).

(111) Erroneous penultimate stress in 3-syllable Light-Light final words because of Parse-syll >> Ft-Bin, Edgemost(R)

/mabada/	*Clash	WSP	Trochee	Non-Fin(σ')	Parse-syll	Ft-Bin	Edgemost(R)	Non-Fin(Ńt)
a. (, ma.ba)('da)				*!		*		*
b. ma('ba.da)					*!		*	*
c. (ma. 'ba)(, da)	*!		*			*	*	
d. ma('ba)(, da)	*!				*	*	*	
e. ('ma.ba)(, da)						*	**	
f. ('ma.ba)da					*!		**	

(112) Erroneous penultimate stress in 3-syllable Heavy-Heavy final words because of Parse-syll >> Ft-Bin, Edgemost(R)

/mabandan/	*Clash	WSP	Trochee	Non-Fin(σ')	Parse-syll	Ft-Bin	Edgemost(R)	Non-Fin(Ńt)
a. (, ma.ban)('dan)		*		*!		*		*
b. ma('ban.dan)		*			*!		*	*
c. (ma. 'ban)(, dan)	*!		*			*	*	
d. ma('ban)(, dan)	*!				*	*	*	
e. ('ma.ban)(, dan)		*				*	**	
f. ('ma.ban)dan		**!			*		**	

4.4.3.3.2 Erroneous antepenultimate stress in XHH in learner with phonotactic stage

As shown in Figure 11 above, the learner with a phonotactic learning stage does not rank Edgemost(R) above Parse-syll. Rather, the ranking of these two constraints

remains undecided (cf. Table 45 below). Whenever the ranking Parse-syll >> Edgemost(R) is chosen, this creates a preference for 3 syllable words to have antepenultimate stress. This is because antepenultimate stress is consistent with all syllables being parsed, while penultimate stress is not. The tableau in (113) shows that antepenultimate stress is indeed obtained for XHH words when Parse-syll >> Edgemost(R).

As can be seen, candidates a., c., d., and f. are ruled out by higher-ranked constraints. The contest between candidates b. (with penultimate stress) and e. (with antepenultimate stress) is decided by the ranking Parse-syll >> Edgemost(R). Candidate b. is ruled out by Parse-syll, even though Edgemost(R) prefers that candidate. Thus, antepenultimate stress, as in candidate e., remains as the winner.

(113) Erroneous antepenultimate stress in XHH words because of Parse-syll >> Edgemost(R)

/mabandan/	*Clash	WSP	Non-Fin(σ)	Trochee	Ft-Bin	Parse-syll	Edgemost(R)	Non-Fin(Ft)
a. (, ma.ban)('dan)		*	*!					*
b. ma('ban.dan)		*				*!	*	*
c. (ma.'ban)(, dan)	*!			*			*	
d. ma('ban)(, dan)	*!					*	*	
e. ('ma.ban)(, dan)		*					**	
f. ('ma.ban)dan		**!				*	**	

However, for XLL words, the fact that Ft-Bin is categorically above Parse-syll makes the mutual ranking of Parse-syll and Edgemost(R) irrelevant to obtaining penultimate stress, as shown in tableau (114). Candidates a., c., and d. are once again ruled out by high-ranked constraints. However, candidate e. now has a foot that is not binary, so that it is excluded by Ft-Bin. Candidate f. is excluded by its excessive violation of Edgemost(R),

so that penultimate stress, as in candidate b., emerges as the winner.

(114) Parse-syll >> Edgemost(R) still generates penultimate stress in XLL words

/mabada/	*Clash	WSP	Non-Fin(ó)	Trochee	Ft-Bin	Parse-syll	Edgemost(R)	Non-Fin(Ft)
a. (,ma.ba)(,da)			*!		*			*
☞ b. ma('ba.da)						*	*	*
c. (ma.'ba)(,da)	*!			*	*		*	
d. ma('ba)(,da)	*!				*	*	*	
e. ('ma.ba)(,da)					*!		**	
f. ('ma.ba)da						*	**!	

XHL words receive penultimate stress because antepenultimate stress is excluded by WSP, as shown in tableau (115). While candidates c. and d. are still excluded by their violation of *Clash, candidates a. (with final stress) and e.-f. (with antepenultimate stress) are not excluded by WSP. Penultimate stress, as in candidate b., is the only remaining option.

(115) Parse-syll >> Edgemost(R) still generates penultimate stress in XHL words

/mabanda/	*Clash	WSP	Non-Fin(ó)	Trochee	Ft-Bin	Parse-syll	Edgemost(R)	Non-Fin(Ft)
a. (,ma.ban)(,da)		*!	*		*			*
☞ b. ma('ban.da)						*!	*	*
c. (ma.'ban)(,da)	*!			*	*		*	
d. ma('ban)(,da)	*!				*	*	*	
e. ('ma.ban)(,da)		*!			*		**	
f. ('ma.ban)da		*!				*	**	

The tableaux in (114) and (115) show that regular penultimate stress in 3-syllable words does not require the ranking Edgemost(R) >> Parse-syll, unless it is in XHH words. Furthermore, in 4-syllable words, Parse-syll actually prefers penultimate stress over

antepenultimate stress, so that the ranking Edgemost(R) >> Parse-syll is not necessary even for Heavy-Heavy final words. As can be seen in tableau (116), candidates a., c., d., f., and g. are excluded by high-ranked constraints. Antepenultimate stress, as in e., is excluded by Parse-syll, and pre-antepenultimate stress, as in h., is excluded by Edgemost(R).

(116) XXHH also retain penultimate stress under Parse-syll >> Edgemost(R)

/rododendrɔn/	*Clash	WSP	Non-Finality(ó)	Trochee	Ft-Bin	Parse-syll	Edgemost(R)	Non-Finality(Ft)
a. ro(,do.dɛn)(,drɔn)		*	*!			*		*
☞ b. (,ro.do)(,dɛn.drɔn)		*					*	*
c. ro(do.,dɛn)(,drɔn)	*!			*		*	*	
d. (,ro.do)(,dɛn)(,drɔn)	*!						*	
e. ro(,do.dɛn)(,drɔn)		*				*!	**	
f. (,ro)(,do.dɛn)(,drɔn)	*!	*			*		**	
g. ro(,do.dɛn)drɔn		**!				**	**	
h. (,ro.do)(,dɛn.drɔn)		*					**!*	

Thus, the possibility of Parse-syll >> Edgemost(R), which leads to antepenultimate stress in words of the type XHH, can only be excluded when the learner is presented with XHH words. Since the learner did not see any words of this type, I conclude that the learner could not have generalized to novel items in a more accurate way than shown in Table 43.

4.4.3.3.3 Full ranking probability tables for both learners

In the tables below, the full set of ranking probabilities for both sample results are given. The rows in each table represent the higher ranked constraint in a pairwise ranking, while the columns represent the lower ranked constraint. Each cell represents the probability in the grammar of the pairwise ranking produced by combining a row and a

column. For instance, the cell at the intersection of the row “*Clash >> ...” and the column “... >> Ft-Bin” represents the probability of *Clash >> Ft-Bin.

Cells containing ranking probabilities above of at least 0.95 are shaded, so that all (near-)categorical rankings are immediately visible. Furthermore, the probabilities of the mutual rankings of Ft-Bin, Parse-syll, and Edgemost(R) are highlighted.

Table 44. Ranking probabilities table for the first run of the simulation without a phonotactic learning stage

	$\begin{smallmatrix} \wedge \\ \wedge \\ \dots \end{smallmatrix}$ *Clash	$\begin{smallmatrix} \wedge \\ \wedge \\ \dots \end{smallmatrix}$ WSP	$\begin{smallmatrix} \wedge \\ \wedge \\ \dots \end{smallmatrix}$ Trochee	$\begin{smallmatrix} \wedge \\ \wedge \\ \dots \end{smallmatrix}$ Non-Fin(ó)	$\begin{smallmatrix} \wedge \\ \wedge \\ \dots \end{smallmatrix}$ Parse-syll	$\begin{smallmatrix} \wedge \\ \wedge \\ \dots \end{smallmatrix}$ Ft-Bin	$\begin{smallmatrix} \wedge \\ \wedge \\ \dots \end{smallmatrix}$ Edgemost(R)	$\begin{smallmatrix} \wedge \\ \wedge \\ \dots \end{smallmatrix}$ Non-Fin(Ĥt)	$p(C \gg \dots)$
*Clash >> ...	-	1.00	0.96	0.99	1.00	1.00	1.00	1.00	0.99
WSP >> ...	0.00	-	0.49	0.97	1.00	0.85	1.00	1.00	0.76
Trochee >> ...	0.04	0.51	-	0.72	0.83	1.00	1.00	1.00	0.73
Non-Fin(ó) >> ...	0.01	0.03	0.28	-	0.67	1.00	1.00	1.00	0.57
Parse-syll >> ...	0.00	0.00	0.17	0.33	-	1.00	1.00	1.00	0.50
Ft-Bin >> ...	0.00	0.15	0.00	0.00	0.00	-	0.85	1.00	0.29
Edgemost(R) >> ...	0.00	0.00	0.00	0.00	0.00	0.15	-	1.00	0.16
Non-Fin(Ĥt) >> ...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-	0.00

Table 45. Ranking probabilities table for the first run of the simulation with a phonotactic learning stage

	$\begin{matrix} \wedge \wedge \\ \dots \\ \text{*Clash} \end{matrix}$	$\begin{matrix} \wedge \wedge \\ \dots \\ \text{WSP} \end{matrix}$	$\begin{matrix} \wedge \wedge \\ \dots \\ \text{Non-Fin}(\acute{o}) \end{matrix}$	$\begin{matrix} \wedge \wedge \\ \dots \\ \text{Trochee} \end{matrix}$	$\begin{matrix} \wedge \wedge \\ \dots \\ \text{Ft-Bin} \end{matrix}$	$\begin{matrix} \wedge \wedge \\ \dots \\ \text{Edgemost(R)} \end{matrix}$	$\begin{matrix} \wedge \wedge \\ \dots \\ \text{Parse-syll} \end{matrix}$	$\begin{matrix} \wedge \wedge \\ \dots \\ \text{Non-Fin}(\acute{f}t) \end{matrix}$	$p(C \gg \dots)$
*Clash >> ...	-	0.84	0.86	0.95	0.99	1.00	1.00	1.00	0.95
WSP >> ...	0.16	-	0.80	0.82	0.75	1.00	1.00	1.00	0.79
Non-Fin(\acute{o}) >> ...	0.14	0.20	-	0.88	0.87	1.00	1.00	1.00	0.73
Trochee >> ...	0.05	0.18	0.12	-	0.58	0.98	0.95	1.00	0.55
Ft-Bin >> ...	0.01	0.25	0.13	0.42	-	1.00	1.00	1.00	0.54
Edgemost(R) >> ...	0.00	0.00	0.00	0.02	0.00	-	0.52	1.00	0.22
Parse-syll >> ...	0.00	0.00	0.00	0.05	0.00	0.48	-	1.00	0.22
Non-Fin($\acute{f}t$) >> ...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-	0.00

As can be seen in Table 44, the rankings Parse-syll >> Ft-Bin and Parse-syllable >> Edgemost(R) are indeed near-categorical with a probability of 1.00. As was reviewed in section 4.4.3.3.1 above, this results in antepenultimate stress being chosen for 3-syllable words by the grammar resulting from the simulation with no phonotactic learning stage.

In Table 45, on the other hand, Ft-Bin >> Parse-syll and Ft-Bin >> Edgemost(R) have a probability of 1.00. Edgemost(R) >> Parse-syll, however, has a probability of 0.52. As reviewed in section 4.4.3.3.2, the possibility of Parse-syll >> Edgemost(R) leads to the possibility of antepenultimate stress on XHH words.

4.4.3.4 Accounting for exceptions: indexed constraints

In this section, I will examine the indexed constraints and the words indexed to them produced at the sample runs whose non-indexed rankings were shown in section 4.4.3.3.

First, in section 4.4.3.4.1, I will show the analysis found by the first iteration of

the learner without a phonotactic learning stage. This is a low-performance result, because that particular run yields a relatively low accuracy on the training data (89.3%), even though some other runs of the learner without a phonotactic learning stage do yield high accuracy on the training data, and a relatively low accuracy on the test data (74.4%), similarly to all other runs of the learner without a phonotactic learning stage. I will show that while this analysis has some flaws, it still finds the same analysis for final stress and penultimate stress in X L H words as in the theoretical analysis in section 4.2.2.2, and also perfectly accounts for all exceptional forms in 4-syllable words.

In section 4.4.3.4.2, I will discuss the first iteration of the learner with a phonotactic learning stage. This is a high-performance result, since it yields both a high accuracy on the training data (95.8%) and a high accuracy on the test data (88.8%). I will show that this analysis is practically identical to the one in section 4.2.2.2, save for some slight redundancy in the set of indexed constraints.

4.4.3.4.1 Indexed constraints found for a low-performance run

For the first run without a phonotactic learning stage, which is a low-performance run because of its low accuracy both on training data and test data, seven indexed constraints were induced, shown in (117) below with the words that they are indexed to:

(117) Indexed constraints induced for low-performance run

a. Non-Fin($\acute{\sigma}$)_{*i*} with *i* on {vabáda, zabáda, jabáda, kabádan} ({kabádan} = all XLH words with penultimate stress)

b. Trochee_{*j*} with *j* on {kalabádan, vabáda} ({kalabádan} = all XXLH words with penultimate stress)

c. Non-Fin($\acute{F}t$)_{*k*} with *k* on {malábada, jalábada} (= all 4-syllable words with antepenultimate stress)

d. Ft-Bin_{*m*} with *m* on {fabáda, vabáda, vabáda, sabáda, zabáda, xabáda, pabánda, fabánda, kabánda}

e. Parse-syll_n with n on all XXLH words with penultimate stress + all 4-syllable words with antepenultimate stress

f. *Clash_p with p on {fabáda, vabáda, sabáda} + all 4-syllable words with antepenultimate stress

g. Edgemost(R)_q with q on {fabáda, vabáda, vabáda, sabáda, zabáda, fabáda} + all words with final stress + all XLH words with penultimate stress + all 4-syllable words with antepenultimate stress

Table 46 summarizes this information. This table shows, for each word type in the training data, to which indexed constraints it belongs. “Non-indexed” means that none of the words of the given type belong to an indexed constraint, while parentheses indicate that only some of the words of the given type are marked for the relevant constraint. Cells marked “-” stand for word types not attested in the training data. As in Table 40 and Table 41, the location of stress according to the QS rule laid out in section 4.2.1 is indicated by shading.

Table 46. Words' partiality to indexed constraints per weight and stress type; low-performance run

Weight pattern	No. of words		
	Antep. stress	Penult stress	Final stress
XXLL	Non-Fin($\acute{F}t$) _k Parse-syll _n *Clash _p	non-indexed	-
XXLH	-	Trochee _j Parse-syll _n	Edgemost(R) _q
XXHL	-	non-indexed	-
XXHH	-	-	-

Weight pattern	No. of words		
	Antep. stress	Penult stress	Final stress
XLL	non-indexed	(Non-Fin($\acute{\sigma}$) _i) (Trochee _j) (Ft-Bin _m) (*Clash _p) (Edgemost(R) _q)	Edgemost(R) _q
XLH	non-indexed	Non-Fin($\acute{\sigma}$) _i Edgemost(R) _q	Edgemost(R) _q
XHL	-	(Ft-Bin _m)	Edgemost(R) _q
XHH	-	-	-

As can be seen in Table 46, this system leaves XXLL and XXHL words with penultimate stress unmarked, as well as XLL and XLH words with antepenultimate stress. This is consistent with the fact that stress in 3-syllable words is antepenultimate in the regular grammars obtained from this type of simulation.

At the same time, some of the XHL words with penultimate stress are marked for Ft-Bin_m, even though this is not necessary: as was shown in Table 40, penultimate stress is predicted by the non-indexed grammar for this word type.

In addition, despite the fact that all XLL words with penultimate stress were marked for some indexed constraint, there is no single indexed constraint that covers all

XLL words with penultimate stress, but, instead, subsets of these words are indexed to every indexed constraint except $\text{Non-Fin}(\acute{\text{Ft}})_k$ and Parse-syll_n (which is logical, since the desired penultimate stress parse, $X(\acute{\text{L}}L)$, violates both of these constraints).

This indicates a certain measure of redundancy. XHL words with penultimate stress need not be marked, since this pattern is already predicted by the non-indexed grammar. Since all XLL words have the same violation pattern, the configuration $\text{Non-Fin}(\acute{\sigma})_j \gg \text{Edgemost(R)}_q \gg \text{Parse-syll}$ would be sufficient to generate penultimate stress for all these words, and indexed versions of Trochee, Ft-Bin, and *Clash are not necessary.

The indexed constraints predicted by the theoretical analysis in section 4.2.2.2 are all found in the analysis. Exceptional final stress is encoded by an indexed version of Edgemost(R) , as in (117g). Exceptional antepenultimate stress (which only exists in XXLL words for the grammar under consideration here) is indeed encoded by an indexed version of $\text{Non-Fin}(\acute{\text{Ft}})$, as in (117c), which is also predicted by the theoretical analysis in section 4.2.2.2. Finally, exceptional penultimate stress in Light-Heavy final words is achieved by an indexed version of $\text{Non-Fin}(\acute{\sigma})$ together with an indexed version of Edgemost(R) for 3-syllable words, as predicted by the analysis and as shown in (117a,g).

However, once again, there is redundancy in the analysis. XXLL words with antepenultimate stress are also indexed to versions of Parse-syll , *Clash, and Edgemost(R) . These latter three constraints do prefer antepenultimate stress over penultimate stress in XXLL words, but they are not necessary when $\text{Non-Fin}(\acute{\text{Ft}})_k$ is sufficiently highly ranked.

Also, XXLH words with penultimate stress are marked for Trochee_j and Parse-

syll_n, as shown in in (117b,e). Parse-syllable and Trochee together prefer penultimate stress over antepenultimate stress in 4-syllable words. Ranking both constraints over WSP means that (,ka.la)(^ˈba.dan) > ka(^ˈla.ba)(,dan) and (,ka.la)(^ˈba.dan) > (ka.^ˈla)(ba.,dan).

Table 47 below shows the probabilities of the indexed constraints in (117) being ranked above the other constraints in the analysis. Constraints are once again ordered by their probability of dominating any other constraint, $p(C \gg \dots)$, and the format is the same as in Table 44 and Table 45 above. Important values that will be discussed below are boxed.

Table 47. Ranking probabilities for all indexed constraints in sample analysis found for low-performance run

	... Non-Fin(^ˈ σ) _i Trochee _j Non-Fin(^ˈ ft) _k *Clash Ft-Bin _m Parse-syll _n *Clash _p Edgemost(R) _q WSP Trochee Non-Fin(^ˈ σ) Parse-syll Ft-Bin Edgemost(R) Non-Fin(^ˈ ft) ...
Non-Fin(^ˈ σ) _i >> ...	-	0.5	0.5	0.99	0.69	0.5	0.5	1	1	1	1	1	1	1	1
Trochee _j >> ...	0.5	-	0.5	0.01	0.28	0.72	0.5	1	1	0.99	1	1	1	1	1
Non-Fin(^ˈ ft) _k >> ...	0.5	0.5	-	0.38	0.5	0.95	0.67	1	0.5	1	1	0.5	1	1	1
Ft-Bin _m >> ...	0.31	0.72	0.5	0.96	-	0.5	0.99	1	0.75	0.03	0.34	1	1	1	1
Parse-syll _n >> ...	0.5	0.28	0.05	0.21	0.5	-	0.25	1	1	1	1	1	1	1	1
*Clash _p >> ...	0.5	0.5	0.33	0.88	0.01	0.75	-	1	0.5	0.05	1	0.99	1	1	1
Edgemost(R) _q >> ...	0	0	0	0.21	0	0	0	-	1	1	1	1	1	1	1

As can be seen in this table, Edgemost(R)_q is dominated by all other indexed constraints.

Edgemost(R)_q itself dominates all non-indexed constraints except *Clash. This ranking is sufficient to derive the exceptional stress patterns, since Edgemost(R)_q is the only indexed constraint that is violated by the desired winners for the words that it is indexed to: exceptional penultimate stress in 3-syllable words requires the ranking Non-Fin(^ˈσ)_i above Edgemost(R)_q, but this is the only ranking among indexed constraints necessary.

For the rest, it is sufficient that all indexed constraints be above WSP.

4.4.3.4.2 Indexed constraints found for a high-performance run

For the grammar resulting from the first run of the learner with a phonotactic learning stage, which is a high-performance because of its high accuracy on both training and test data, the following indexed constraints were found:

(118) Indexed constraints induced for high-performance run

- a. Non-Fin($\acute{\sigma}$)_i with *i* on {kabádan, kalabádan} (all XLH penult stress words)
- b. Edgemost(R)_j, with *j* on {malábada, jalábada, kabádan, kalabádan, sabandá, nabadá, rabadá, fabadán, sabadán, xabadán, xalabadán} (all 4-syllable antepenult stress words, all XLH penult stress words, all final stress words)
- c. Edgemost(R)_{j,k}, with *k* on {kabádan, kalabádan, sabandá, nabadá, rabadá, fabadán, sabadán, xabadán} (all XLH penult stress words, all 3-syllable final stress words)
- d. Edgemost(R)_{j,k,m}, with *m* on {kabádan, kalabádan, sabandá} (all XLH penult stress words, all XHL final stress words)
- e. Non-Fin($\acute{F}t$)_n, with *n* on {malábada, jalábada, kabádan, sabandá, rabadá, sabadán} (all 4-syllable antepenult stress words, all 3-syllable XLH penult stress words, some 3-syllable final stress words)
- f. Non-Fin($\acute{F}t$)_{n,o}, with *o* on {malábada, jalábada} (all 4-syllable antepenult stress words)
- g. Parse-syll_p, with *p* on {lábada, jábada, mábada, pábada, tábada, kábada, malábada, jalábada, kalabádan, nabadá, rabadá, sabadán, xabadán} (all antepenult stress words, all 4-syllable XLH penult stress words, some 3-syllable final stress words)

Table 48 shows which type of words are indexed to which constraints, and which types remain unmarked. As in Table 46 in section 4.4.3.4.1 above, “non-indexed” means that none of the words of the given type belong to an indexed constraint, while parentheses indicate that only some of the words of the given type are marked for the relevant constraint. Cells marked “-” stand for word types not attested in the training data, and the location of stress according to the QS rule laid out in section 4.2.1 is indicated by shading.

Table 48. Words' partiality to indexed constraints per weight and stress type; high-performance run

Weight pattern	No. of words			Weight pattern	No. of words		
	Antep. stress	Penult stress	Final stress		Antep. stress	Penult stress	Final stress
X X L L	Edge most(R) _j Non-Fin(Ft) _n Non-Fin(Ft) _{n,o} Parse-syll _p	non-indexed	-	X L L	Parse-syll _p	non-indexed	Edge most(R) _j Edge most(R) _{j,k} (Non-Fin(Ft) _n) (Parse-syll _p)
X X L H	-	Non-Fin(ó) _i Edge most(R) _j Edge most(R) _{j,k} Edge most(R) _{j,k,m} Parse-syll _p	Edge most(R) _j	X L H	non-indexed	Non-Fin(ó) _i Edge most(R) _j Edge most(R) _{j,k} Edge most(R) _{j,k,m} Non-Fin(Ft) _n	Edge most(R) _j Edge most(R) _{j,k}
X X H L	-	non-indexed	-	X H L	-	non-indexed	Edge most(R) _j Edge most(R) _{j,k} Edge most(R) _{j,k,m}
X X H H	-	-	-	X H H	-	-	-

As can be seen in Table 48, there is a perfect separation between rule-obeying and exceptional forms: none of the forms that obey the rule in (66) are marked for any indexed constraint, and for each class of exceptions, all members of that class are indexed to at least one indexed constraint.

As predicted by the theoretical analysis, all words with final stress are indexed to at least one version of Edgemost(R). As will be shown in Figure 13 below, the difference

between the three indexed versions of Edgemost(R) is that Edgemost(R)_j is only ranked above Non-Fin(σ'), whereas the other two versions, Edgemost(R)_{j,k} and Edgemost(R)_{j,k,m}, are also ranked above WSP (making it possible to account for final stress in XHL and XXHL words). There is no distinguishable difference in ranking between the two latter constraints, as can be verified in Table 49.

Exceptional penultimate stress in XLH and XXLH words is accounted for by ranking Non-Fin(σ')_i above Edgemost(R)_{j,k} and Edgemost(R)_{j,k,m}, which themselves are above WSP: Table 48 shows that these forms are indexed to these constraints, while the Hasse diagram in Figure 13 below shows the ranking. This indexation and ranking is precisely as in the theoretical account summarized in (91) in section 4.2.2.2.4.

Finally, exceptional antepenultimate stress in 4-syllable (XXLL) words is accounted for by ranking Non-Fin(σ')_{n,o} above Edgemost(R)_{j,k} and Edgemost(R)_{j,k,m}, as in the theoretical account in (91).

However, 3-syllable (XLL) words with exceptional antepenultimate stress are accounted for by marking them for Parse-syll_p, which is ranked above WSP and Ft-Bin, but below *Clash. The ranking *Clash >> Parse-syll_p, Non-Fin(σ') >> Ft-Bin leads to antepenultimate stress in XLL words, as shown in the tableau in (119): final stress candidate a. is ruled out by Non-Fin(σ'), penultimate stress candidates c. and d. are ruled out by *Clash, while penultimate stress candidate b. is ruled out by Parse-syll_p. Candidate f. is also ruled out by Parse-syll_p, so that candidate e., with penultimate stress, wins.

(119) Antepenultimate stress for /mabada_p/, indexed to Parse-syll_p

/mabada _p /	*Clash	Parse-syll _p	Non-Fin(σ)	Ft-Bin
a. (,ma.ba)(,da)			*!	*
b. ma('ba.da)		*!		
c. (ma.'ba)(,da)	*!			*
d. ma('ba)(,da)	*!	*		*
e. ('ma.ba)(,da)				*
f. ('ma.ba)da		*!		

The Hasse diagram in Figure 13 shows the ranking of the indexed constraints – as in section 4.4.4.2, solid lines represent rankings that have a probability of at least 0.95, while dotted lines represent rankings with a probability between 0.80 and 0.95. The full set of ranking probabilities that underlie the Hasse diagram in Figure 13 are shown in Table 49.

The indexed constraint Non-Fin(Ĥt)_n is at the very bottom of the hierarchy (as can be seen in Table 49, it is dominated by every other constraint other than Non-Fin(Ĥt)), having practically no effect on the outcome of the grammar, and it is therefore not included in the Hasse diagram in Figure 13.

Figure 13. Hasse diagram for analysis found for high-performance run

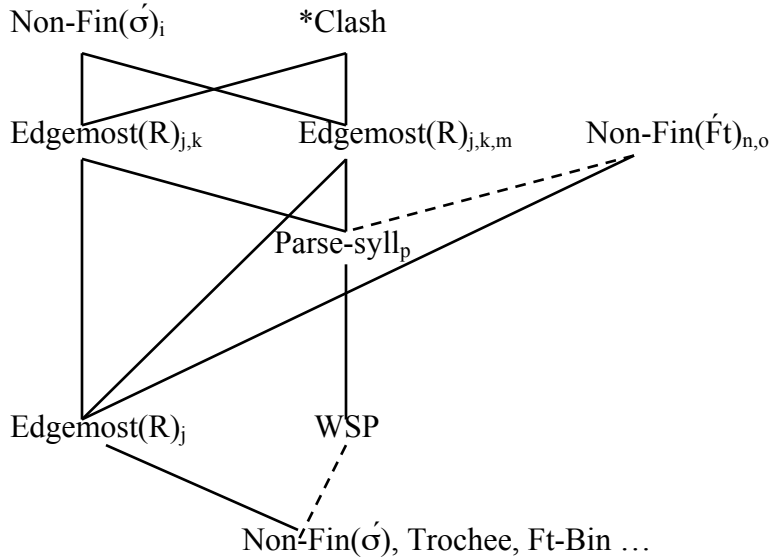


Table 49. Ranking probabilities for all indexed constraints in sample analysis found for high-performance run

	Non-Fin($\acute{\sigma}$) _i ... ^ ^ ...	*Clash ^ ...	Edgemost(R) _{j,k,m} ... ^ ^ ...	Edgemost(R) _{j,k} ... ^ ^ ...	Non-Fin($\acute{F}t$) _{n,o} ... ^ ^ ...	Parse-syll _p ... ^ ^ ...	WSP ^ ^ ...	Edgemost(R) _j ... ^ ^ ...	Non-Fin($\acute{\sigma}$) ... ^ ^ ...	Trochee ^ ^ ...	Ft-Bin ^ ^ ...	Parse-syll ... ^ ^ ...	Edgemost(R) ... ^ ^ ...	Non-Fin($\acute{F}t$) ... ^ ^ ...	Non-Fin($\acute{F}t$) _n ... ^ ^ ...
Non-Fin($\acute{\sigma}$) _i >> ...	-	0.74	1	1	0.5	1	1	1	1	1	1	1	1	1	1
Edgem.(R) _{j,k,m} >> ...	0	0.02	-	0.49	0.5	0.99	0.56	0.96	1	0.99	1	1	1	1	1
Edgem.(R) _{j,k} >> ...	0	0.02	0.51	-	0.5	0.99	0.43	0.96	1	1	1	1	1	1	1
Non-Fin($\acute{F}t$) _{n,o} >> ...	0.5	0.43	0.5	0.5	-	0.11	0.5	1	0.96	1	1	1	1	0.91	0.94
Parse-syll _p >> ...	0	0.01	0.01	0.01	0.89	-	1	0.41	0.36	0.83	1	0.79	1	1	1
Edgem.(R) _j >> ...	0	0	0.04	0.04	0	0.59	0.73	-	0.99	0.94	0.99	0.98	0.98	1	1
Non-Fin($\acute{F}t$) _n >> ...	0	0	0	0	0.06	0	0	0	0	0	0.01	0.01	0	0.44	-

Summarizing, the results of this run yield an indexed constraint analysis almost identical to the one predicted in section 4.2.2.2. There is a slight degree of redundancy in the system. First, instead of one indexed version of Edgemost(R), three are made. Second, instead of one indexed version of Non-Fin($\acute{F}t$), two are made, and the first of these, Non-Fin($\acute{F}t$)_n, is indexed to many forms for which Non-Fin($\acute{F}t$) is not helpful, so that it sinks to the bottom of the hierarchy.

Finally, antepenultimate stress in XLL and XXLL words could be marked by the same indexed constraint (Non-Fin($\acute{F}t$)_{n,o}), but instead, only XXLL words with antepenultimate stress are marked for this constraint, whereas XLL words with antepenultimate stress are marked for Parse-syll_p.

However, unlike the low-performance run discussed in section 4.4.3.4.1, all rule-obeying words are left unmarked for indexed constraints, and for every class of exceptions (XLL words with antepenultimate stress, XHL words with final stress, ...), there is at least one indexed constraints for which the entire class is marked that regulates

that class' behavior in the grammar.

4.4.3.5 Summary of results

As was shown in the preceding parts of section 4.4.3, the learner with a phonotactic learning stage performed very well on all three relevant parameters: coverage of the learning data, generalization to unseen data, and marking of exceptions for indexed constraints. This learner reached an average accuracy of 88.8% on the training data, and produced the expected regular stress pattern summarized in (66) with 88.9% accuracy – a number lessened by imperfect performance on XHH words. In section 4.4.3.3.1, I argue that the lessened accuracy on XHH words stems from the fact that these words were not provided to the learner, and therefore the ranking $\text{Edgemost(R)} \gg \text{Parse-syll}$ necessary only for XHH words could not be learned. In other words, the learner simply could not do better on XHH words.

The learner without a phonotactic learning stage performed similarly in terms of accuracy on the training data (92%). The difference with the pattern in (66) is that, on the test set of novel words, XLL and XHH words were assigned antepenultimate rather than penultimate stress: 4-syllable words followed the pattern in (66), while 3-syllable words did not – which leads to a mere 74% accuracy on the desired pattern summarized in (66). I showed that this stems from too high a ranking of Parse-syll . The failure to discover a single pattern for 3-syllable and 4-syllable words could be caused by the lack of a phonotactic learning stage in which all data are evaluated together. This shows that the learner presented in this chapter is effective, but performs less efficiently on hidden structure (identifying exceptions) when it is not given a phonotactic learning stage, which is an additional argument in favor of Jarosz's (2006a) two-staged approach to learning

phonotactic generalizations.

When analyzing the constraints used to represent exceptions, I found that a high-performance run (one with high accuracy on both training and test data) showed a perfect division between rule-obeying and exceptional forms in marking words for exceptional constraints. A small group of constraints was picked for indexation, and the analysis was very similar to that given in the theoretical analysis, summarized in (91) in section 4.2.2.2.4.

A low-performance run (one with low accuracy on both training and test data) showed a less precise and efficient analysis. However, it was still able to extract a very good analysis for 4-syllable words, X L H words, and final stress words.

4.5 Conclusion

In this chapter, I presented a learner that can represent lexical exceptions to a lexicon-wide grammar by inducing lexically indexed constraints (Pater 2000, 2010) in a probabilistic OT framework. Previous approaches to indexed constraint induction (Becker 2007, 2008, 2009, Coetzee 2009b, Pater 2010) use Recursive Constraint Demotion (Tesar 1995) as a learning framework. This framework is categorical rather than probabilistic, i.e., it constructs an ordering of constraints that is not variable and is not sensitive to quantitative patterns in the data, which is problematic in at least two ways. First, Jarosz (2013a) has shown that hidden structure in general (with foot structure as a case study) is learned more efficiently when a probabilistic approach is taken. Second, probabilistic approaches are able to deal with within-word variation (see, for instance Coetzee and Pater 2011). If a learner of lexical indexation, which is a type of hidden structure and models between-word variation, is to be combined with learners that

learn other types of hidden structure as well as within-word variation, a probabilistic formulation is preferred.

The learner is cast in the Expectation Driven Learning framework (EDL; Jarosz submitted), which employs Expectation Maximization for constraint rankings. EDL defines grammars as sets of probabilities over constraint rankings (see section 4.3.1.1), and learning proceeds by considering every pair of constraints in a tableau separately, and evaluating which ranking is more likely to derive each data point given the current grammar; new probabilities over constraint rankings are derived from these values (see section 4.3.1.3 for a more detailed description). Probabilities over constraint rankings can be computed both given the entire dataset, and given an individual word.

After each pass through the data, which produces new constraint ranking probabilities, each individual word's ranking probability for each constraint pair is compared to the lexicon-wide ranking probability for that constraint pair. If, given a constraint pair $\{A, B\}$, a word's probability of $B \gg A$ and the lexicon-wide probability of $A \gg B$ both exceed a threshold (in this case, 60%), that word is considered deviant for that constraint pair (see (103) in section 4.3.2).

The constraint pair whose deviant words have the highest summed deviation from the lexicon-wide ranking probability for that constraint pair is used as the basis of inducing or updating an indexed constraint. In other words, a lexically indexed constraint is induced if it is a member of the “most exceptional” constraint pair. This induction or updating of indexed constraints is done after every pass of the learner through the data; see section 4.3.2 for a detailed description of the lexically indexed constraint induction procedure.

Two learners were set up: one with a phonotactic learning stage (Hayes 2004, Prince and Tesar 2004, Jarosz 2006a), and one without. The phonotactic learning stage entails that no indexed constraints are induced for the initial period of learning, to prevent indexed constraints from explaining away patterns that are true of the entire lexicon in terms of word-specific properties. This possibility was given to the learner with a phonotactic learning stage to maximize its chances of arriving at the correct default pattern.

These two learners were tested on the problem of Dutch main stress assignment (van der Hulst 1984, Kager 1989, Nouveau 1994, van Oostendorp 1997, 2012, Gussenhoven 2014). Dutch main stress is governed by a quantity-sensitive rule (as confirmed by experimental work, summarized in section 4.2.1.3), but there is also widespread exceptionality, so that antepenultimate, penultimate, and final stress are attested in all combinations of Heavy and Light syllables. The task of the learner is to find a lexicon-wide grammar that encodes the QS stress rule and accounts for the exceptions.

The learners were trained on a dataset based on the set of 3 and 4-syllable monomorphemes without schwa or superheavy syllables in Dutch (as reported by Ernestus and Neijt 2008). The learner with a phonotactic learning stage was very successful: the learner reached the 95% accuracy criterion, generalized to novel items according to the QS rule proposed in the literature and confirmed by experimental work (except for XHH words, which I argue in section 4.4.3.3.2 is the result of XHH words not being a part of the data set), and a set of indexed constraints that covers only and all the exceptions is induced.

The learner without a phonotactic learning stage was slightly less successful: it was equally successful on 4-syllable forms, but it predicted antepenultimate stress on XLL and XHH forms because of too high a ranking for Parse-syllable. Its coverage of exceptions for 3-syllable words was also imperfect. This diminished accuracy of the learner without a phonotactic learning stage argues for the necessity of a phonotactic learning stage.

Future work is needed to corroborate and further explore the result obtained here. First and foremost, the model should be applied to other complex cases of exceptionality, in particular, cases where segmental processes are at work. Cases that exhibit both exceptionality (between-word variation) and within-word variation are also very important to consider in this framework, since one of the advantages of probabilistic OT frameworks is that they can account for within-word variation (see also the beginning of section 4.1).

Another important direction for future work is to connect the default grammar (as derived by the learner) to the way in which speakers' judgments on non-words are influenced by exceptions, for instance as reported in section 4.1.1.3. Building on literature like Becker (2007, 2008, 2009), Becker and Fainleib (2009), and Linzen, Kasyanenko, and Gouskova (2013), Gouskova, Newlin-Łukowicz, and Kasyanenko (2015), Becker and Gouskova (2016), it can be hypothesized that novel words are stochastically assigned indices that make them exceptional. However, it is possible that it is not only the number of words that carry an exceptional index and the similarity of a nonce word to existing words that carry this index that are important, but also the perceived origin of a word (see the comments on this in section 4.1.1.3). More data on

this are necessary, and a model of stochastically generalizing indexed constraints to novel items should be developed.

A comparison between the constraint-splitting approach taken by Becker (2009) and Coetzee (2009b), and the stringency approach from Pater (2000) (see section 4.1.1) taken here should also be undertaken in the future – which can be done with slight adjustments to the indexed constraint induction procedure described in section 4.3.2.

Finally, since the main innovation proposed in this chapter is the extension of indexed constraint induction to a probabilistic framework, an explicit comparison of the current model to the categorical models developed by Becker (2009), Coetzee (2009b), and Pater (2010) should be performed. Another important thing to consider is the issue of “splitting” versus “lumping” in finding lexical indices (“splitting”: start with one index for all exceptions and refine as needed; “lumping”: start with separate indices for each exception and lump them together as needed; see section 4.1.1). The work done by Moore-Cantwell and Pater (to appear) forms a middle ground between these approaches: it shows that an approach that starts with separate indices for each word (not just each exceptional word) can account for both rule-obeying and exceptional forms, and can also derive the relative strength of exceptional stress in a language; however, it does not tackle the explicit induction of indices. Regardless of these issues, the fact that lexically indexed constraints can be induced in a probabilistic OT framework is an important result in itself.

APPENDIX A. ONE-SEGMENT PHONOLOGICAL PATTERNS AS FOUND IN MIELKE (2007)

This appendix gives an overview of one-segment patterns found in Mielke's (2007) P-base database, and consists of two parts. The first part documents all patterns that are encoded in P-base as targeting one segment. The second part is the result of a manual search through a subset of the database (namely, all languages starting in A) for generalizations which are encoded as targeting all segments but one.

Part 1. Patterns coded as targeting one segment

Total number of patterns: 13

1. Abha Arabic:

/ʔ/ → glide / X__a: (and a: __V)
X may only be [a:] (maybe also [u]).

2. Cherokee (Oklahoma dialect)

X → voiced / __l (/d/ then becomes /l/)
X may only be [t].

3. Kinnauri

X → [w] / __[a]
X may only be [u] (maybe also [o,ɔ]).

4. Kanakuru

/t/ (or /ɬ/?) → [m] / __X
X may only be [n] (maybe also [ɲ]).

5. Malayalam

vls unaspl stops → voiced / X__
X may only be [ŋ] (maybe also [m,ɳ,n,ɳn]).

6. Malayalam

C1 of NC clusters may only be [ŋ] (maybe also [m,ɳ,n,ɳn]).

7. Mupun (Jipari dialect)

$X \rightarrow \text{voiced} / V_V$

X may only be [p] (maybe also [tʃ,k,f,s]).

8. Mokilese

glide inserted / $X_ \{i, u, e, o, \varepsilon, \text{ɔ}\}, \{i, u, e, o, \varepsilon, \text{ɔ}\}_ X$

X may only be [a].

9. Oromo, Harar (Eastern Oromo)

$X \rightarrow \text{voiceless} / C_V\#$ (V is devoiced too)

X may only be [ɕ] (maybe also [dʰ,b,dʒ,g]).

10. Purik

C2 of initial CC cluster when C1 is a nasal may only be [j] (maybe also [w]).

11. Purik

C3 of medial CCC cluster when C1 is lateral or trill and C2 is a stop may only be [j] (maybe also [w]).

13. Tauya

V1 of verb stem-final VV may only be [u].

14. Wolio

Only [k] may be replaced by corresponding prenasalized stop at certain morpheme boundaries (maybe also [p,t,c,b,dʒ,g]).

Part 2. Manual search among the set of languages starting with A: segment classes which include “all but one”

Number of patterns: 11

1. Auyana;

1p sg-dl-pl $\rightarrow [si-] / _+X$

2p-3p dl-pl $\rightarrow [ti-] / _+X$

X may be any consonant but [w] ([ʔ] “maybe” participates in this pattern) – the allomorphs [su-] and [tu-] appear when X is [w], instead.

2. Asmat (Flamingo Bay Dialect)

/tʃ/ → [t] / __+X

X may be any consonant but [ɾ].

3. Asmat (Flamingo Bay Dialect)

X+X → X (*i.e., degemination of two identical consonants at a morpheme boundary – A.N.*)

X may be any consonant but [ɾ].

4. Akan

Any consonant but [ŋ] may occur in stem-initial position.

5. Anywa (aka Anuak)

Any consonant but [w] undergoes plosivization (in the modified noun formation) (/w/ geminates to [ww] instead).

6. Afar

Any vowel but [a(:)] becomes close next to its corresponding glide.

7. Afar

Most verbs starting in any vowel but [a(:)] take prefixes.

8. Afar

Only [i(:)] undergoes full assimilation to a neighboring vowel separated from it by a word edge:

[i(:)] → V_x / _#V_x , V_x#_

9. Amharic

/t/ (imperfect, jussive, verbal noun) → __+X

X may not be ɡ

10. Azari, Iranian (South Azerbaijani)

Among consonants, [ʔ] may not occur initially.

11. Arabic, Moroccan

Among consonants, [ʔ] may not be geminate.

**APPENDIX B. FULL RESULTS FOR FEATURE LEARNING SIMULATIONS
DESCRIBED IN CHAPTER 2**

Run #	Constraints (features encoded as sets { })	Weights
1	*#{mnŋ}	3.87
	*m#	2.45
	*{iu}{pm}{iu}	2.26
	*{iu}b{iu}	1.57
	*{iu}{pbm}	0.08
	*{pbm}{iu}	0.05
2	*#{mnŋ}	3.93
	*{iu}{pb}{iu}	2.16
	*m#	2.15
	*{iu}{pbm}u	0.74
	*{iu}m	0.72
3	*#{mnŋ}	3.85
	*m#	2.47
	*{iu}{pb}{iu}	2.39
	*{iu}m{iu}	1.67
	*m{aiu}	0.03
4	*#{mnŋ}	3.96
	*{iu}{pbm}{iu}	2.90
	*m#	2.57
5	*#{mnŋ}	3.96
	*{iu}{pbm}{iu}	2.88
	*m#	2.56
	*{iu}{pbm}	0.02
6	*{iu}{pbm}{iu}	2.88
	*m#	2.56
	*#{mnŋ}{aiu}	1.98
	*#{mnŋ}	1.98
	*{iu}{pbm}	0.02
7	*#{mnŋ}	3.87
	*m#	2.36
	*{iu}{pbm}i	1.50
	*{iu}{pbm}u	1.50
	*{iu}{pb}{iu}	0.91
	*{iu}m	0.21
8	*#{mnŋ}	3.86
	*{aiu}m#	2.47
	*{iu}{pm}{iu}	2.39
	*{iu}b{iu}	1.70

9	*#{mnŋ}	3.96
	*{iu}{pbm}{iu}	2.90
	*m#	2.57
10	*#{mnŋ}	3.96
	*{iu}{pbm}{iu}	2.90
	*m#	2.57
11	*{iu}{pbm}{iu}	2.72
	*#m	2.68
	*m#	2.36
	*#{nŋ}	1.12
	*#{nŋ}{aiu}	1.12
	*#{nŋ}	1.12
	*{aiu}m	0.04
12	*#{mnŋ}	3.79
	*{iu}{pb}{iu}	2.41
	*m#	2.29
	*{iu}m{iu}	1.57
	*{aiu}m	0.14
	*#{mnŋ}{aiu}	0.08
	*{aiu}m#	0.05
13	*{iu}{pbm}{iu}	2.90
	*{aiu}m#	2.57
	*#{mnŋ}	1.32
	*#{mnŋ}{aiu}	1.32
	*#{mnŋ}	1.32
14	*#{mnŋ}	3.85
	*{aiu}m#	2.47
	*{iu}{pb}{iu}	2.39
	*{iu}m{iu}	1.67
	*m{aiu}	0.03
15	*#{mnŋ}	3.85
	*m#	2.47
	*{iu}{pb}{iu}	2.39
	*{iu}m{iu}	1.67
	*m{aiu}	0.03
16	*#{mŋ}	2.78
	*#{nŋ}	2.78
	*{iu}{pbm}{iu}	2.77
	*m#	2.27
	*{m}#	0.17
17	*#{nŋ}	3.37
	*{iu}{pbm}{iu}	2.73
	*#m	2.68
	*m#	2.15
	*{aiu}m#	0.25

18	*#{mnŋ}	3.94
	*{iu}{pb}{iu}	2.17
	*m#	2.09
	*u{pbm}{iu}	0.74
	*{iu}m	0.62
	*{aiu}m	0.11
19	*#{mnŋ}	3.86
	*m#	2.48
	*{iu}{pm}{iu}	2.37
	*{iu}b{iu}	1.67
	*{iu}{pb}	0.03
	*{iu}{pb}u	-1.39E-17
20	*#{mnŋ}	3.96
	*{iu}{pbm}{iu}	2.90
	*m#	2.57
21	*#{mnŋ}	3.96
	*{iu}{pbm}{iu}	2.88
	*m#	2.56
	*{iu}{pbm}	0.02
22	*#{mnŋ}	3.96
	*{iu}{pbm}{iu}	2.90
	*m#	2.57
23	*#{mnŋ}	3.96
	*{iu}{pbm}{iu}	2.90
	*{m}#	2.57
24	*#{mnŋ}	3.87
	*{iu}{pb}{iu}	2.40
	*{aiu}m#	2.36
	*{iu}m{iu}	1.50
	*{iu}m	0.21
25	*#{mnŋ}	3.85
	*m#	2.47
	*{iu}{pb}{iu}	2.39
	*{iu}m{iu}	1.67
	*m{aiu}	0.03
26	*#{mnŋ}	3.96
	*{iu}{pbm}{iu}	2.90
	*m#	2.57
	*{pbm}{iu}	0.00

27	*#{mnŋ}	3.91
	*{iu}{pb}{iu}	2.36
	*m#	2.14
	*{iu}m	0.73
	*u{pm}{iu}	0.44
	*mi	0.09
	*mu	0.09
28	*#{mnŋ}	3.96
	*{iu}{pbm}{iu}	2.90
	*{aiu}m#	2.57
29	*#{mnŋ}	3.87
	*m#	2.34
	*{iu}{pbm}u	1.57
	*{iu}{pbm}i	1.57
	*{iu}{pb}{iu}	0.84
	*{aiu}m	0.14
30	*#{mnŋ}	3.96
	*{iu}{pbm}{iu}	2.90
	*m#	2.57
31	*#{mnŋ}	3.96
	*{iu}{pbm}{iu}	2.90
	*m#	2.57
32	*#{mnŋ}	3.86
	*m#	2.47
	*{iu}{pb}{iu}	2.39
	*{iu}m{iu}	1.70

APPENDIX C. TRAINING DATA FOR THE SIMULATIONS DESCRIBED IN CHAPTER 4

Table 50. List of all words given to the learner with their main stress patterns

4-syllable words		3-syllable words	
XXLL	malábada jalábada talabáda kalabáda	XLL	pábada mábada tábada lábada jábada kábada fabáda vabáda vabáda sabáda zabáda jabáda χabáda nabadá rabadá
XXLH	kalabádan χalabadún	XLH	pábadan mábadan tábadan lábadan jábadan kabádan fabadón sabadón χabadón
XXHL	malabánda	XHL	pabánda fabánda tabánda kabánda sabandá

**APPENDIX D. ALL METRICAL PARSES CONSIDERED IN THE
SIMULATIONS DESCRIBED IN CHAPTER 4**

Table 51. List of all parses considered by the learner

4-syllable words		3-syllable words	
Pre-antepenultimate stress	(' σ σ) (, σ σ)		
	(' σ) (, σ σ) (, σ)		
	(' σ) (, σ σ) σ		
	(' σ) (σ , σ) (, σ)		
	(' σ) (σ , σ) σ		
Antepenultimate stress	(σ ' σ) (σ , σ)	Antepenultimate stress	(' σ) (, σ σ)
	(, σ) (' σ σ) (, σ)		(' σ) (σ , σ)
	(, σ) (' σ σ) σ		(' σ σ) (, σ)
	σ (' σ σ) (, σ)		(' σ σ) σ
	σ (' σ σ) σ		
Penultimate stress	(, σ σ) (' σ σ)	Penultimate stress	(, σ) (' σ σ)
	(, σ) (σ ' σ) (, σ)		σ (' σ σ)
	(, σ) (σ ' σ) σ		(σ ' σ) (, σ)
	σ (σ ' σ) (, σ)		(σ ' σ) σ
	σ (σ ' σ) σ		
Final stress	(σ , σ) (σ ' σ)	Final stress	(, σ) (σ ' σ)
	(, σ) (, σ σ) (' σ)		σ (σ ' σ)
	σ (, σ σ) (' σ)		(, σ σ) (' σ)
	(, σ) (σ , σ) (' σ)		(σ , σ) (' σ)
	σ (σ , σ) (' σ)		

BIBLIOGRAPHY

- Ahoua, Firmin. 2009. "Areal Typology of Tone-Consonant Interaction and Implosives in Kwa, Kru, and Southern-Mande." In *Form and Function in Language Research*, edited by Johannes Helmbrecht, Yoko Nishina, Yong-Min Shin, Stavros Skopeteas, and Elisabeth Verhoeven, 217–30. Berlin: Mouton de Gruyter.
- Akers, Crystal. 2011. "Commitment-Based Learning of Hidden Linguistic Structures." PhD dissertation, Rutgers University.
- Albright, Adam, and Bruce Hayes. 2002. "Modeling English Past Tense Intuitions with Minimal Generalization." In *SIGPHON 6: Proceedings of the Sixth Meeting of the ACL Special Interest Group in Computational Phonology*, 58–69. ACL.
- . 2003. "Rules vs. Analogy in English Past Tenses: A Computational/experimental Study." *Cognition* 90: 119–61. doi:10.1016/S0010-0277(03)00146-X.
- Alderete, John, Adrian Brasoveanu, Nazarré Merchant, Alan S. Prince, and Bruce B. Tesar. 2005. "Contrast Analysis Aids the Learning of Phonological Underlying Forms." In *Proceedings of the 24th West Coast Conference on Formal Linguistics*, edited by John Alderete, Chung-hye Han, and Alexei Kochetov, 34–42. Somerville, MA: Cascadia Proceedings Project.
- Allen, Blake, and Michael Becker. under revision. "Learning Alternations from Surface Forms with Sublexical Phonology"
- Anderson, Stephen R. 1981. "Why Phonology Isn't Natural." *Linguistic Inquiry* 12: 493–539.
- Anttila, Arto. 1997. "Deriving Variation from Grammar." In *Variation, Change, and Phonological Theory*, edited by Frans Hinskens, Roeland van Hout, and W. Leo Wetzels, 35–68. Amsterdam: John Benjamins.
- . 2002. "Morphologically Conditioned Phonological Alternations." *Natural Language & Linguistic Theory* 20 (1): 1–42.
- Apoussidou, Diana. 2007. "The Learnability of Metrical Phonology." PhD dissertation, University of Amsterdam.
- Archangeli, Diana, Jeff Mielke, and Douglas Pulleyblank. 2012. "From Sequence Frequencies to Conditions in Bantu Vowel Harmony: Building a Grammar from the Ground Up." In *Phonological Explorations: Empirical, Theoretical and Diachronic Issues*, edited by Bert Botma and Roland Noske, 191–222. Berlin: Mouton de Gruyter.

- Ashlock, Daniel. 2006. *Evolutionary Computation for Modeling and Optimization*. New York: Springer.
- Avery, Peter, and Keren Rice. 1989. "Constraining Underspecification." In *Proceedings of the Northeast Linguistic Society* 19, 1–15. Amherst, MA: GLSA.
- Baayen, Harald, Richard Piepenbrock, and Leon Gulikers. 1995. *The CELEX Lexical Database (CD-ROM)*. Philadelphia, PA: Linguistics Data Consortium, University of Pennsylvania.
- Becker, Michael. 2007. "From the Lexicon to a Stochastic Grammar." presented at the 38th Meeting of the North East Linguistic Society, University of Ottawa.
- . 2008. "The Role of Markedness Constraints in Learning Lexical Trends." presented at the 82nd Annual Meeting of the Linguistic Society of America, Chicago, IL.
- . 2009. "Phonological Trends in the Lexicon: The Role of Constraints." PhD dissertation, University of Massachusetts, Amherst.
- Becker, Michael, and Lena Fainleib. 2009. "The Role of Markedness Constraints in Learning Lexical Trends." presented at the 83rd Annual Meeting of the Linguistic Society of America, San Francisco, CA.
- Becker, Michael, and Maria Gouskova. 2016. "Source-Oriented Generalizations as Grammar Inference in Russian Vowel Deletion." *Linguistic Inquiry* 47 (3).
- Beckman, Mary E., and Jan R. Edwards. 1990. "Lengthenings and Shortenings and the Nature of Prosodic Constituency." In *Papers in Laboratory Phonology. Volume 1, Between the Grammar and Physics of Speech*, edited by John Kingston and Mary E. Beckman, 152–78. Cambridge: Cambridge University Press.
- Berger, Adam L., Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. "A Maximum Entropy Approach to Natural Language Processing." *Computational Linguistics* 22: 39–71.
- Berko, Jean. 1958. "The Child's Learning of English Morphology." *Word* 14: 150–77.
- Bermúdez-Otero, Ricardo. 1999. "Constraint Interaction in Language Change [Opacity and Globality in Phonological Change]." PhD dissertation, University of Manchester & Universidad de Santiago de Compostela.
- . 2003. "The Acquisition of Phonological Opacity." In *Variation within Optimality Theory: Proceedings of the Stockholm Workshop on Variation within Optimality Theory*, 25–36.

- Blevins, Juliette. 2004. *Evolutionary Phonology: The Emergence of Sound Patterns*. Cambridge: Cambridge University Press.
- . 2006. “A Theoretical Synopsis of Evolutionary Phonology.” *Theoretical Linguistics* 32 (2): 117–66.
- Boersma, Paul. 1998. “Functional Phonology: Formalizing the Interactions between Articulatory and Perceptual Drives.” PhD dissertation, University of Amsterdam.
- . 2007. “Some Listener-Oriented Accounts of H-Aspiré in French.” *Lingua* 117 (12): 1989–2054.
- . 2011. “A Programme for Bidirectional Phonology and Phonetics and Their Acquisition and Evolution.” In *Bidirectional Optimality Theory*, edited by Anton Benz and Jason Mattausch. Amsterdam: John Benjamins.
- Boersma, Paul, and Kateřina Chládková. 2013. “Detecting Categorical Perception in Continuous Discrimination Data.” *Speech Communication* 55: 33–39.
- Boersma, Paul, and Paola Escudero. 2008. “Learning to Perceive a Smaller L2 Vowel Inventory: An Optimality Theory Account.” In *Contrast in Phonology: Theory, Perception, Acquisition*, edited by Peter Avery, B. Elan Dresher, and Keren Rice, 271–301. Berlin: Mouton de Gruyter.
- 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 J. McCarthy and Joe Pater. London: Equinox Press.
- Boersma, Paul, and Jan-Willem van Leussen. 2015. “Efficient Evaluation and Learning in Multi-level Parallel Constraint Grammars.” Manuscript. University of Amsterdam.
- Booij, Geert. 1995. *The Phonology of Dutch*. Oxford: Clarendon Press.
- Boyeldieu, Pascal. 1985. *La Language Lua (‘niellim’): (Groupe Boua - Moyen-Chari, Tchad) Phonologie - Morphologie - Dérivation Verbale*. Cambridge: Cambridge University Press.
- Brown, Roger, and Camille Hanlon. 1970. “Derivational Complexity and Order of Acquisition in Child Speech.” In *Cognition and the Development of Language*, edited by John R. Hayes, 11–53. New York: Wiley.
- Bybee, Joan. 2001. *Phonology and Language Use*. Cambridge: Cambridge University Press.

- Bye, Patrik. 2006. "Grade Alternation in Inari Saami and Abstract Declarative Phonology." Manuscript. Universitetet i Tromsø.
- Byrd, Richard H., Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 1995. "A Limited Memory Algorithm for Bound Constrained Optimization." *SIAM Journal on Scientific Computing* 16 (5): 1190–1208. doi:10.1137/0916069.
- Calamaro, Shira, and Gaja Jarosz. 2015. "Learning General Phonological Rules from Distributional Information: A Computational Model." *Cognitive Science* 39 (3): 647–66.
- Chládková, Kateřina. 2014. "Finding Phonological Features in Perception." PhD dissertation, University of Amsterdam.
- Chomsky, Noam. 1964. *Current Issues in Linguistic Theory*. The Hague: Mouton.
- Chomsky, Noam, and Morris Halle. 1968. *The Sound Pattern of English*. New York, Evanston, and London: Harper and Row.
- Chomsky, Noam, Morris Halle, and Fred Lukoff. 1956. "On Accent and Juncture in English." In *For Roman Jakobson: Essays on the Occasion of His Sixtieth Birthday*, edited by Morris Halle, Horace Lunt, Hugh Mclean, and Cornelis H. Van Schooneveld, 65–80. The Hague: Mouton.
- Clements, George N. 2000. "Phonology." In *African Languages: An Introduction*, edited by Bernd Heine and Derek Nurse, 123–60. Cambridge: Cambridge University Press.
- Clements, George N. 2003. "Feature Economy in Sound Systems." *Phonology* 20 (3): 287–333.
- Clements, George N., and Elizabeth Hume. 1995. "The Internal Organization of Speech Sounds." In *Handbook of Phonological Theory*, edited by John A. Goldsmith, 245–306. Oxford: Blackwell.
- Coetzee, Andries W. 2009a. "An Integrated Grammatical/non-Grammatical Model of Phonological Variation." In *Current Issues in Linguistic Interfaces*, edited by Hye-Kyung Kang, Young-Se Kang, Jong-Yurl Yoon, Hyunkyung Yoo, Sze-Wing Tang, Yong-Soon Kang, Youngjun Jang, Chul Kim, Kyoung-Ae Kim, 2:267–94. Seoul: Hankookmunhwasa.
- . 2009b. "Learning Lexical Indexation." *Phonology* 26 (1): 109–45.

- Coetzee, Andries W, and Joe Pater. 2011. "The Place of Variation in Phonological Theory." In *Handbook of Phonological Theory*, edited by John A. Goldsmith, Jason Riggle, and Alan C. Yu, 2nd ed., 401–34. Wiley-Blackwell.
- Colavin, Rebecca S., Roger Levy, and Sharon Rose. 2010. "Modeling OCP-Place in Amharic with the Maximum Entropy Phonotactic Learner." Manuscript. University of California, San Diego. <http://idiom.ucsd.edu/~rose/Modeling%20OCP.pdf>.
- Comrie, Bernard. 1967. "Irregular Stress in Polish and Macedonian." *International Review of Slavic Linguistics* 1: 227–40.
- Cristià, Alejandrina, Jeff Mielke, Robert Daland, and Sharon Peperkamp. 2013. "Constrained Generalization of Implicitly Learned Sound Patterns." *Journal of Laboratory Phonology* 4 (2): 259–85.
- Cristià, Alejandrina, and Amanda Seidl. 2008. "Is Infants' Learning of Sound Patterns Constrained by Phonological Features?" *Language Learning and Development* 4 (3): 203–27.
- De Lacy, Paul, and John Kingston. 2013. "Synchronic Explanation." *Natural Language and Linguistic Theory* 31 (2): 287–355.
- Della Pietra, Stephen A., Vincent J. Della Pietra, and John D. Lafferty. 1997. "Inducing Features of Random Fields." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19: 380–93.
- Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society. Series B (Methodological)* 39 (1): 1–38.
- Dillon, Brian, Ewan Dunbar, and William J Idsardi. 2013. "A Single Stage Approach to Learning Phonological Categories: Insights from Inuktitut." *Cognitive Science* 37 (2): 344–77.
- Dogil, Grzegorz. 1988. "Phonological Configurations: Natural Classes, Sonority and Syllabicity." In *Features, Segmental Structure and Harmony Processes*, edited by Harry van der Hulst and Norval Smith, 1:79–103. Dordrecht: Foris.
- 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*, 1–38.
- Dresher, B. Elan. 2009. *The Contrastive Hierarchy in Phonology*. Cambridge: Cambridge University Press.

- . 2013. “The Arch Not the Stones: Universal Feature Theory without Universal Features.” presented at the Conference on Features in Phonology, Morphology, Syntax and Semantics: What are they?, Center for Advanced Study in Theoretical Linguistics (CASTL), University of Tromsø.
- . 2014. “The Arch Not the Stones: Universal Feature Theory without Universal Features.” *Nordlyd* 41 (2): 165–81.
- Dresher, B. Elan, and Jonathan D. Kaye. 1990. “A Computational Learning Model for Metrical Phonology.” *Cognition* 34: 137–95.
- Eisenstat, Sarah. 2009. “Learning Underlying Forms with MaxEnt.” MA thesis, Brown University.
- Elman, Jeffrey L., and David Zipser. 1988. “Learning the Hidden Structure of Speech.” *Journal of the Acoustical Society of America* 83 (4): 1615–26.
- Elsner, Micha, Stephanie Antetomaso, and Naomi H Feldman. 2016. “Joint Word Segmentation and Phonetic Category Induction.” In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 59–65.
- Elsner, Micha, Sharon Goldwater, Naomi H Feldman, and Frank Wood. 2013. “A Joint Learning Model of Word Segmentation, Lexical Acquisition, and Phonetic Variability.” In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 42–54.
- Emeneau, M. B. 1961. *Kolami: A Dravidian Language*. Annamalainagar: Annamalai University.
- Ernestus, Mirjam, and Anneke Neijt. 2008. “Word Length and the Location of Primary Word Stress in Dutch, German, and English.” *Linguistics* 46 (507-540).
- Escudero, Paola, and Paul Boersma. 2003. “Modelling the Perceptual Development of Phonological Contrasts with Optimality Theory and the Gradual Learning Algorithm.” In *Proceedings of the 25th Annual Penn Linguistics Colloquium*, edited by Sudha Arunachalam, Elsi Kaiser, and Alexander Williams, 1:71–85. Penn Working Papers in Linguistics 8. Philadelphia, PA: Department of Linguistics, University of Pennsylvania.
- . 2004. “Bridging the Gap Between L2 Speech Perception Research and Phonological Theory.” *Studies in Second Language Acquisition* 26: 551–85.
- Everitt, Brian N., Sabine Landau, Morven Leese, and Daniel Stahl. 2011. *Cluster Analysis*. 5th edition. Wiley Series in Probability and Statistics. Chichester, West Sussex: Wiley.

- Eychenne, Lucien. 2014. "Schwa and the Loi de Position in Southern French." *Journal of French Language Studies* 24 (2): 223–53.
- Fraley, Chris, Adrian E. Raftery, and Luca Scrucca. 2012. "Mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation." Technical Report N. 597. Department of Statistics, University of Washington.
- Gallagher, Gillian. 2014. "Evidence for an Identity Bias in Phonotactics." *Laboratory Phonology* 5 (3): 337–78.
- Goldrick, Matthew A. 2001. "Turbid Output Representations and the Unity of Opacity." In *Proceedings of the Northeast Linguistic Society* 30, Rutgers University, edited by Masako Hirotani, Andries Coetzee, Nancy Hall, and Ji-Yung Kim, 231–45. Amherst, MA: GLSA.
- Goldsmith, John A. 1976. "Autosegmental Phonology." PhD dissertation, MIT.
- Goldsmith, John A., and Aris Xanthos. 2009. "Learning Phonological Categories." *Language* 85 (1): 4–38.
- Goldwater, Sharon, and Mark Johnson. 2003. "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, 111–20.
- Gouskova, Maria, Luiza Newlin-Lukowicz, and Sofya Kasyanenko. 2015. "Selectional Restrictions as Phonotactics over Sublexicons." *Lingua* 167: 41–81.
- Gussenhoven, Carlos. 2009. "Vowel Duration, Syllable Quantity, and Stress in Dutch." In *The Nature of the Word. Essays in Honor of Paul Kiparsky*, edited by Kristin Hanson and Sharon Inkelas, 181–98. Cambridge, MA: MIT Press.
- . 2014. "Possible and Impossible Exceptions in Dutch Word Stress." In *Word Stress: Theoretical and Typological Issues*, edited by Harry van der Hulst, 276–96. Cambridge: Cambridge University Press.
- Guy, Gregory R. 2011. "Variability." In *The Blackwell Companion to Phonology*, edited by Marc van Oostendorp, Colin J. Ewen, Elizabeth Hume, and Keren Rice, 2190–2213. Oxford: Blackwell.
- Halle, Morris. 1959. *The Sound Pattern of Russian*. The Hague: Mouton.
- . 1978. "Knowledge Unlearned and Untaught: What Speakers Know about the Sounds of Their Language." In *Linguistic Theory and Psychological Reality*, edited by Morris Halle, Joan Bresnan, and George A. Miller, 294–303. Cambridge, Massachusetts: MIT Press.

- Hall, Tracy A., and Marzena Żygiś. 2010. "An Overview of the Phonology of Obstruents." In *Turbulent Sounds. An Interdisciplinary Guide*, edited by Susanne Fuchs, Martine Toda, and Marzena Żygiś, 1–37. Berlin: Mouton de Gruyter.
- Hammond, Michael. 1999. *The Phonology of English: A Prosodic Optimality-Theoretic Approach: A Prosodic Optimality-Theoretic Approach*. Oxford: Oxford University Press.
- Hayes, Bruce. 1995. *Metrical Stress Theory: Principles and Case Studies*. Chicago, IL: University of Chicago Press.
- . 1999. "Phonetically Driven Phonology: The Role of Optimality Theory and Inductive Grounding." In *Functionalism and Formalism in Linguistics, Vol. I: General Papers*, edited by Michael Darnell, Edith Moravcsik, Michael Noonan, Frederick J. Newmeyer, and Kathleen Wheatly, 243–85. Amsterdam: John Benjamins.
- . 2004. "Phonological Acquisition in Optimality Theory: The Early Stages." In *Fixing Priorities: Constraints in Phonological Acquisition*, edited by René Kager, Joe Pater, and Wim Zonneveld, 158–203. Cambridge, Massachusetts: Cambridge University Press.
- Hayes, Bruce, and Colin Wilson. 2008. "A Maximum Entropy Model of Phonotactics and Phonotactic Learning." *Linguistic Inquiry* 39: 379–440.
- Heinz, Jeffrey. 2009. "On the Role of Locality in Learning Stress Patterns." *Phonology* 26 (2): 303–51.
- Heinz, Jeffrey, and Cesar Koirala. 2010. "Maximum Likelihood Estimation of Feature-Based Distributions." In *Proceedings of the 11th Meeting of the ACL-SIGMORPHON, ACL 2010*, 28–37.
- Hoerl, Arthur E., and Robert W. Kennard. 1970. "Ridge Regression: Biased Estimation for Nonorthogonal Problems." *Technometrics* 12 (1): 55–67.
- 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 (2): 151–95.
- Inkelas, Sharon. 1998. "The Theoretical Status of Morphologically Conditioned Phonology: A Case Study from Dominance." *Yearbook of Morphology* 1997: 121–55.
- Inkelas, Sharon, and C. Orhan Orgun. 1995. "Level Ordering and Economy in the Lexical Phonology of Turkish." *Language* 71: 763–93.

- . 1998. “Level (Non)ordering in Recursive Morphology: Evidence from Trukish.” In *Morphology and Its Relation to Phonology and Syntax*, edited by Steven Lapointe, Diane Brentari, and Patrick Farrell, 360–92. Stanford, CA: CSLI Publications.
- Inkelas, Sharon, and Cheryl Zoll. 2007. “Is Grammar Dependence Real? A Comparison Between Cophonological and Indexed Constraint Approaches to Morphologically Conditioned Phonology.” *Linguistics* 45 (1): 133–71.
- Itô, Junko, and Armin Mester. 1995. “The Core-Periphery Structure of the Lexicon and Constraints on Reranking.” In *University of Massachusetts Occasional Papers in Linguistics*, edited by Jill Beckman, Suzanne Urbanczyk, and Laura Walsh, Vol. 18: *Papers in Optimality Theory*:181–209. Amherst, MA: GLSA.
- . 1999. “The Structure of the Phonological Lexicon.” In *The Handbook of Japanese Linguistics*, edited by Natsuko Tsujimura, 62–100. Oxford: Blackwell.
- Jarosz, Gaja. submitted. “Expectation Driven Learning of Phonology”
- . 2006a. “Rich Lexicons and Restrictive Grammars: Maximum Likelihood Learning in Optimality Theory.” PhD dissertation, Johns Hopkins University.
- . 2006b. “Richness of the Base and Probabilistic Unsupervised Learning in Optimality Theory.” In *Proceedings of the 8th Meeting of the ACL Special Interest Group in Computational Phonology*, edited by Richard Wicentowski and Grzegorz Kondark, 50–59. New York: Association for Computational Linguistics.
- . 2013a. “Learning with Hidden Structure in Optimality Theory and Harmonic Grammar: Beyond Robust Interpretive Parsing.” *Phonology* 30 (01): 27–71.
- . 2013b. “Naïve Parameter Learning for Optimality Theory - The Hidden Structure Problem.” In *NELS 40: Proceedings of the 40th Annual Meeting of the North East Linguistic Society*, edited by Seda Kan, Claire Moore-Cantwell, and Robert Staubs, 2:1–14. Amherst, MA: GLSA.
- . 2015. “Learning Opaque and Transparent Interactions in Harmonic Serialism.” presented at the 2015 Annual Meeting on Phonology, Vancouver, BC.
- Jensen, John T. 1993. *English Phonology*. John Benjamins.
- Jesse, Alexandra, James M. McQueen, and Mike Page. 2007. “The Locus of Talker-Specific Effects in Spoken-Word Recognition.” In *Proceedings of ICPHS XVI*, 1921–24.

- Johnson, Mark. 1984. "A Discovery Procedure for Certain Phonological Rules." In 10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics, 344–47.
- . 1992. "Identifying a Rule's Context from Data."
- Johnson, Mark, Joe Pater, Robert Staubs, and Emmanuel Dupoux. 2015. "Sign Constraints on Feature Weights Improve a Joint Model of Word Segmentation and Phonology." In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 303–13. Denver, Colorado: Association for Computational Linguistics. <http://www.aclweb.org/anthology/N15-1034>.
- Joos, Martin. 1942. "A Phonological Dilemma in Canadian English." *Language*, 141–44.
- Kager, René. 1989. *A Metrical Theory of Stress and Destressing in English and Dutch*. Dordrecht - Holland/Providence RI - USA: Foris.
- Kaye, Jonathan. 1990. "What Ever Happened to Dialect B?" In *Grammar in Progress: GLOW Essays for Henk van Riemsdijk*, edited by Joan Mascaró and Marina Nespor. Dordrecht: Foris, 259–63.
- Kenstowicz, Michael, and Charles Kisseberth. 1979. *Generative Phonology: Description and Theory*. Orlando, FL: Academic Press.
- Kim, Heejin, and Jennifer Cole. 2005. "The Stress Foot as a Unit of Planned Timing: Evidence from Shortening in the Prosodic Phrase." In *INTERSPEECH-2005*, 2365–68.
- Kimper, Wendell. 2011. "Positive Constraints and Finite Goodness in Harmonic Serialism." Manuscript. University of Massachusetts Amherst.
- Kiparsky, Paul. 1971. "Historical Linguistics." In *A Survey of Linguistic Science*, edited by William O. Dingwall. College Park, Maryland: University of Maryland Press.
- . 1973. "Abstractness, Opacity and Global Rules." In *Three Dimensions of Linguistic Theory*, edited by Osamu Fujimura, 57–86. Tokyo: TEC.
- . 2000. "Opacity and Cyclicity." *The Linguistic Review* 17 (2-4): 351–66.
- . 2006. "The Amphichronic Program vs. Evolutionary Phonology." *Theoretical Linguistics* 32 (3): 217–36.

- Kirby, Simon, and James R. Hurford. 2002. "The Emergence of Linguistic Structure: An Overview of the Iterated Learning Model." In *Simulating the Evolution of Language*, edited by Angelo Cangelosi and Domenico Parisi, 121–47. New York: Springer.
- Kraska-Szlenk, Iwona. 1995. "The Phonology of Stress in Polish." PhD dissertation, University of Illinois, Urbana-Champaign.
- Kullback, S., and R. A. Leibler. 1951. "On Information and Sufficiency." *Ann. Math. Statist.* 22 (1): 79–86.
- Kuo, Li-Jen. 2009. "The Role of Natural Class Features in the Acquisition of Phonotactic Regularities." *Journal of Psycholinguistic Research* 38: 129–50.
- Lieberman, Mark, and Alan Prince. 1977. "On Stress and Linguistic Rhythm." *Linguistic Inquiry* 8 (2): 249–336.
- Lin, Ying. 2005. "Learning Features and Segments from Waveforms: A Statistical Model of Early Phonological Acquisition." PhD dissertation, University of California, Los Angeles.
- Lin, Ying, and Jeff Mielke. 2008. "Discovering Place and Manner Features: What Can Be Learned from Acoustic and Articulatory Data?" In *Penn Working Papers in Linguistics* 14, edited by Joshua Tauberer, Aviad Eliaim, and Laurel MacKenzie, 1:241–54.
- Linzen, Tal, and Gillian Gallagher. under review. "Rapid Generalization in Phonotactic Learning"
- . 2014. "The Timecourse of Generalization in Phonotactic Learning." In *Proceedings of Phonology 2013*, edited by John Kingston, Claire Moore-Cantwell, Joe Pater, and Robert Staubs. Washington, DC: Linguistic Society of America.
- Linzen, Tal, Sofya Kasyanenko, and Maria Gouskova. 2013. "Lexical and Phonological Variation in Russian Prepositions." *Phonology* 30 (3): 453–515.
- Maddieson, Ian, and Kristin Precoda. 1990. "Updating UPSID." In *UCLA Working Papers in Phonetics* 74:, 104–11. Department of Linguistics, University of California, Los Angeles.
- Manning, Christopher D., and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: MIT Press.
- Marcus, G.F. 1993. "Negative Evidence in Language Acquisition." *Cognition* 46: 53–85.

- McCarthy, John J. 1979. "Formal Problems in Semitic Phonology and Morphology." PhD dissertation, Massachusetts Institute of Technology.
- . 1981a. "A Prosodic Theory of Nonconcatenative Morphology." *Linguistic Inquiry* 12 (3): 373–418.
- . 1981b. "The Role of the Evaluation Metric in the Acquisition of Phonology." In *The Logical Problem of Language Acquisition*, edited by C. L. Baker and John J. McCarthy. Cambridge, Massachusetts: MIT Press.
- McCarthy, John J. 2008. "The Gradual Path to Cluster Simplification." *Phonology* 25: 271–319.
- McMurray, Bob, Richard N. Aslin, and Joseph C. Toscano. 2009. "Statistical Learning of Phonetic Categories: Insights from a Computational Approach." *Developmental Science* 12 (3): 369–78.
- McQueen, James M., Anne Cutler, and Dennis Norris. 2006. "Phonological Abstraction in the Mental Lexicon." *Cognitive Science* 30: 1113–26.
- Melvold, Janis. 1989. "Structure and Stress in the Phonology of Russian." PhD dissertation, Massachusetts Institute of Technology.
- Merchant, Nazarré. 2008. "Discovering Underlying Forms: Contrast Pairs and Ranking." PhD dissertation, Rutgers University.
- Merchant, Nazarré, and Bruce B. Tesar. 2008. "Learning Underlying Forms by Searching Restricted Lexical Subspaces." In *Proceedings of the Forty-First Conferences of the Chicago Linguistics Society*, 2:141–63. Malden, MA: Wiley-Blackwell.
- Mielke, Jeff. 2004. "The Emergence of Distinctive Features." PhD dissertation, Ohio State University.
- . 2007. P-Base, Version 1.92. University of Ottawa.
- Moore-Cantwell, Claire, and Joe Pater. to appear. "Gradient Exceptionality in Maximum Entropy Grammar with Lexically Specific Constraints." *Catalan Journal of Linguistics* 15.
- Morén, Bruce. 2006. "Consonant-Vowel Interactions in Serbian: Features, Representations and Constraint Interactions." *Lingua* 116 (8): 1198–1244.
- Moreux, Bernard. 1985. "La Loi de Position En Français Du Midi. I. Synchronie (Béarn)." *Cahiers de Grammaire* 9: 45–138.

- . 2006. “Les Voyelles Moyennes En Français Du Midi: Une Tentative de Synthèse En 1985.” *Cahiers de Grammaire* 30: 307–17.
- Nedjalkov, Igor. 1997. *Evenki*. New York: Routledge.
- Newport, Elissa L. 2016. “Statistical Language Learning: Computational, Maturational and Linguistic Constraints.” *Language and Cognition* 8 (3): 447–61.
- Nielsen, Kuniko. 2011. “Specificity and Abstractness in VOT Imitation.” *Journal of Phonetics* 39: 132–42.
- Niyogi, Partha. 2004. “Towards a Computational Model of Human Speech Perception.” In *Proceedings of the Conference on Sound to Sense, MIT (In Honor of Ken Stevens’ 80th Birthday)*, 208–22.
- Nouveau, Dominique. 1994. *Language Acquisition, Metrical Theory, and Optimality. A Study of Dutch Word Stress*. Utrecht: OTS Publications.
- Odden, David. 2011. “Rules v. Constraints.” In *The Handbook of Phonological Theory*, 1–39. Wiley-Blackwell. <http://dx.doi.org/10.1002/9781444343069.ch1>.
- Orgun, C. Orhan. 1996. “Sign-Based Morphology and Phonology: With Special Attention to Optimality Theory.” PhD dissertation, University of California, Berkeley.
- Pater, Joe. 2000. “Non-Uniformity in English Secondary Stress: The Role of Ranked and Lexically Specific Constraints.” *Phonology* 17 (2): 237–74.
- . 2009. “Weighted Constraints in Generative Linguistics.” *Cognitive Science* 33: 999–1035.
- . 2010. “Morpheme-Specific Phonology: Constraint Indexation and Inconsistency Resolution.” In *Phonological Argumentation: Essays on Evidence and Motivation*, edited by Steve Parker, 123–54. London: Equinox Press.
- . 2012. “Emergent Systemic Simplicity (and Complexity).” In *Proceedings from Phonology in the 21st Century: In Honour of Glyne Piggott*, edited by Jenny Loughran and Alanah McKillen. Vol. 1. McGill Working Papers in Linguistics 22. Montreal: McGill University.
- . 2014. “Canadian Raising with Language-Specific Weighted Constraints.” *Language* 90 (1): 230–40.
- . 2016. “Universal Grammar with Weighted Constraints.” In *Harmonic Grammar and Harmonic Serialism*, edited by John J McCarthy and Joe Pater. London: Equinox Press.

- Pater, Joe, Karen Jesney, Robert Staubs, and Brian Smith. 2012. "Learning Probabilities over Underlying Representations." In *Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology*, 62–71. Association for Computational Linguistics.
- Peperkamp, Sharon, Rozenn Le Calvez, Jean-Pierre Nadal, and Emmanuel Dupoux. 2006. "The Acquisition of Allophonic Rules: Statistical Learning with Linguistic Constraints." *Cognition* 101: B31–41.
- Pierrehumbert, Janet. 2003a. "Phonetic Diversity, Statistical Learning, and Acquisition of Phonology." *Language and Speech* 46 (2): 115–54.
- . 2003b. "Probabilistic Phonology: Discrimination and Robustness." In *Probability Theory in Linguistics*, edited by Rens Bod, Jennifer Hay, and Stefanie Jannedy, 177–228. Cambridge, Massachusetts: MIT Press.
- Pinker, Steven, and Alan S Prince. 1988. "On Language and Connectionism: Analysis of a Parallel Distributed Processing Model of Language Acquisition." *Cognition* 28 (1): 73–193.
- Pizzo, Presley. 2013. "Learning Phonological Alternations with Online Constraint Induction." presented at the Old Word Conference in Phonology, Boğaziçi University, İstanbul.
- 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): 1–41.
- Prickett, Brandon. in progress. "Complexity as a Pressure Against Generalization." University of Massachusetts Amherst.
- Prince, Alan S. 1997. "Stringency and Anti-Paninian Hierarchies." presented at the LSA Institute. <http://rucss.rutgers.edu/images/personal-alan-prince/gamma/talks/insthdt2.pdf>.
- . 2003. "Anything Goes." In *A New Century of Phonology and Phonological theory: a Festschrift for Professor Shosuke Haraguchi on the Occasion of his Sixtieth Birthday*, edited by Takeru Honma, Masao Okazaki, Toshiyuki Tabata, and Shin-ichi Tanaka, 66–90. Tokyo: Kaitakusha.
- . 2007. "The Pursuit of Theory." In *Cambridge Handbook of Phonology*, edited by Paul De Lacy, 22–46. Cambridge: Cambridge University Press.
- Prince, Alan S., and Paul Smolensky. 1993. *Optimality Theory: Constraint Interaction in Generative Grammar*. Malden, MA: Blackwell.

- Prince, Alan S., and Bruce B. Tesar. 2004. "Learning Phonotactic Distributions." In *Fixing Priorities: Constraints in Phonological Acquisition*, edited by René Kager, Joe Pater, and Wim Zonneveld, 245–91. Cambridge, Massachusetts: Cambridge University Press.
- Quattoni, Ariadna, Sybor Wang, Louis-Philippe Morency, and Trevor Darrell. 2007. "Hidden Conditional Random Fields." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (10): 1848–53.
- Rasin, Ezer, and Roni Katzir. 2016. "On Evaluation Metrics in Optimality Theory." *Linguistic Inquiry* 47 (2): 235–82.
- R Core Team. 2013. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <http://www.R-project.org/>.
- Reetz, Henning. 1999. "Web Interface to UPSID." http://web.phonetik.uni-frankfurt.de/upsid_info.html.
- Rizzolo, Olivier. 2002. "Du Leurre Phonétique Des Voyelles Moyennes En Français et Du Divorce Entre Licenciement et Licenciement Pour Gouverner." PhD dissertation, Université de Nice-Sophia Antipolis.
- Saffran, Jenny R., and Erik D. Thiessen. 2003. "Pattern Induction by Infant Language Learners." *Developmental Psychology* 39 (3): 484–94.
- Selkirk, Elizabeth. 1978. "The French Foot: On the Statute of 'mute' E." *Studies in French Linguistics* 2: 79–141.
- Shipley, William J. 1964. *Maidu Grammar*. Berkeley, CA: University of California Press.
- Smolensky, Paul, and Geraldine Legendre. 2006. *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar*. Cambridge, Massachusetts: MIT Press.
- Soderstrom, Melanie, Donald Mathis, and Paul Smolensky. 2006. "Abstract Genomic Encoding of Universal Grammar in Optimality Theory." In *The Harmonic Mind*, by Paul Smolensky and Geraldine Legendre, 403–71. Cambridge, Massachusetts: MIT Press.
- Stankiewicz, Edward. 1993. *The Accentual Patterns of the Slavic Languages*. Stanford, CA: Stanford University Press.
- Staubs, Robert. 2014a. "Computational Modeling of Learning Biases in Stress Typology." PhD dissertation, University of Massachusetts, Amherst.
- . 2014b. *Stratal MaxEnt Solver*.

- Staub, Robert, and Joe Pater. 2016. "Learning Serial Constraint-Based Grammars." In *Harmonic Grammar and Harmonic Serialism*, edited by John McCarthy and Joe Pater. London: Equinox Press.
- Tesar, Bruce B. 1995. "Computational Optimality Theory." University of Colorado.
- . 1998. "An Iterative Strategy for Language Learning." *Lingua* 104: 131–45.
- . 2004. "Using Inconsistency Detection to Overcome Structural Ambiguity." *Linguistic Inquiry* 35: 219–53.
- . 2006. "Faithful Contrastive Features in Learning." *Cognitive Science* 30: 863–903.
- . 2008. "Output-Driven Maps." Manuscript. Rutgers University. ROA-956. Rutgers Optimality Archive.
- . 2011. "Learning Phonological Grammars for Output-Driven Maps." In *Proceedings of the 39th North East Linguistic Society*, edited by Suzi Lima, Kevin Mullin, and Brian Smith, 2:785–898. Amherst, MA: GLSA.
- Tesar, Bruce B., John Alderete, Graham Horwood, and Nazarré Merchant. 2003. "Surgery in Language Learning." In *Proceedings of the 22nd West Coast Conference on Formal Linguistics*, edited by Gina Garding and Mimura Tsujimura, 477–90. Somerville, MA: Cascadia Press.
- Tesar, Bruce B., and Paul Smolensky. 2000. *Learnability in Optimality Theory*. Cambridge, Massachusetts: MIT Press.
- Timberlake, Alan. 2004. *A Reference Grammar of Russian*. Cambridge: Cambridge University Press.
- Trommelen, Mieke, and Wim Zonneveld. 1989. *Klemtoon En Metrische Fonologie* [=Stress and Metrical Phonology]. Muiderberg: Dick Coutinho.
- Vallabha, Gautam K., James L. McClelland, Ferran Pons, Janet F. Werker, and Shigeaki Amano. 2007. "Unsupervised Learning of Vowel Categories from Infant-Directed Speech." *Proceedings of the National Academy of Sciences* 104 (33): 13273–78.
- van der Hulst, Harry. 1984. *Syllable Structure and Stress in Dutch*. Linguistic Models 8. Dordrecht: Foris Publications.
- van Oostendorp, Marc. 1997. "Lexicale Variatie in Optimaliteitstheorie." *Nederlandse Taalkunde* 2: 133–54.
- . 2008. "Incomplete Devoicing in Formal Phonology." *Lingua* 118: 1362–74.

- . 2012. “Quantity and the Three-Syllable Window in Dutch Stress.” *Language and Linguistics Compass* 6 (6): 343–58. doi:10.1002/lnc3.339.
- Ward, Dennis. 1975. “Unaccented Vowels in Russian.” *Russian Linguistics* 2: 91–104.
- Wedel, Andy. 2003. “Self-Organization and Categorical Behavior in Phonology.” *Proceedings of the Berkeley Linguistics Society* 29: 611–22.
- . 2011. “Self-Organization in Phonology.” In *The Blackwell Companion to Phonology*, edited by Marc van Oostendorp, Colin J. Ewen, Elizabeth Hume, and Keren Rice, 1:130–47. Oxford: Blackwell.
- Wilson, Colin. 2010. “Searching for Phonological Generalizations.” presented at the Cornell Workshop on Grammar Induction, Cornell University, Ithaca, NY.
- Wolf, Matthew. 2011. “Exceptionality.” In *The Blackwell Companion to Phonology*, edited by Marc van Oostendorp, Colin J. Ewen, Elizabeth Hume, and Keren Rice, 2538–59. Oxford: Blackwell.
- Wonnacott, Elizabeth, and Elissa L. Newport. 2005. “Novelty and Regularization: The Effect of Novel Instances on Rule Formation.” In *BUCLD 29: Proceedings of the 29th Annual Boston University Conference on Language Development*, edited by Alejna Brugos, Manuella R. Clark-Cotton, and Seungwan Ha, 663–73. Somerville, MA: Cascadilla Press.