The Lester and Sally Entin Faculty of Humanities

School of Philosophy, Linguistics and Science Studies

The Program of Cognitive Studies of Language and Its Uses

An MDL-based computational model for unsupervised joint learning of morphophonological constraints and lexicons in Optimality Theory

MA thesis submitted by

Victoria Costa

Prepared under the guidance of:

Dr. Roni Katzir

December 2017

אוניברסיטת תל אביב
TEL AVIV UNIVERSITY

הפקולטה למדעי הרוח ע"ש לסטר וסאלי אנטין
בית הספר לפילוסופיה, בלשנות ולימודי מדע
התכנית ללימודים קוגנטיבים של השפה ושימושיה

מודל חישובי מבוסס אורך תיאור מינימלי (MDL) ללמידה משותפת בלתי מונחית של אילוצים מורפופונולוגיים ולקסיקונים בתאוריית האופטימליות

חיבור זה הוגש כעבודת גמר לקראת התואר
''מוסמך אוניברסיטה'' – M.A. באוניברסיטת ת''א

על ידי
ויקטוריה קוסטה

העבודה הוכנה בהדרכת:
ד''ר רוני קציר

דצמבר 2017
טבת תשע''ח

תקציר

בעבודה מוצג מודל חישובי ללמידה של דקדוקי תאוריית האופטימליות (OT) שמסוגל להסיק חוקים מורפו-פונולוגיים מ-surface forms שלא עברו ניתוח , וללמוד במשולב פונולוגיה ומורפו-פונולוגיה. המודל משמש כהרחבה למודל חישובי ללמידה של פונולוגיה מבוססת תאוריית האופטימליות שהגו Rasin and Katzir (2016), שמשתמש במטריקה שמבוססת על אורך תיאור מינמלי (MDL). פונקציית המטרה הזו ממזערת את אורך הקידוד של הדיקדוק ואת הקלט בהנתן הדיקדוק ובאתו הזמן מאזנת בין grammar economy ל-grammar restrictiveness. המודל הלומד מקבל כקלט surface forms שלא עברו ניתוח, והוא לומד את ה-(URs) underlying representations ביחד עם היררכייה נכונה של אילוצי נאמנות ומסומננות על ידי חיפוש במרחב ההיפוטזות אינסופי של אילוצי תאוריית אופטימליות אוניברסליים ולקסיקון עם URs.

Abstract

I present a computational model for learning Optimality Theory grammars, which is able to induce morphophonological rules from unanalyzed surface forms and jointly learn phonology and morphophonology. This model serves as an extension of the OT phonology learner proposed by Rasin and Katzir (2016), which uses the Minimum Description Length (MDL) -based evaluation metric, the objective function of which is to minimize the encoding length of grammar and the input data given grammar together, while balancing between grammar economy and restrictiveness. The learner proposed in this work is presented with unanalyzed surface forms and induces the underlying representations (URs) along with the optimal Faithfulness and Markedness constraint hierarchy by searching an infinite hypothesis space of the universal OT constraints and a lexicon with URs.

Acknowledgements

First and foremost, I want to thank my father, Vilen V. Belyi, for introducing me to the world of linguistics through teaching me foreign languages since I was a child and always inspiring me to question everything. He has always been an inspiration to me as a scientist and as a human being.

I would like to express my sincere gratitude to my advisor, Roni Katzir, for his guidance and encouragement in my research and in writing this thesis. His immense knowledge and engaging courses made me fascinated with the computational side of linguistics and cognition in general, which led me to discover and explore the field of machine learning. Without his advice on acquiring the necessary knowledge related to computer science and my research, this thesis would not have been possible.

I am grateful to Evan Cohen and Outi Bat-El for their advice, insights and recommendations, and for suggesting alternative ways of looking into the issues.

My sincere thanks also goes to Iddo Berger, whose vast programming knowledge helped me with every step of my work on the learning model and advanced my Python skills as well. I also wish to thank Nur Lan, Netanel Haim and Ezer

Rasin for their help and feedback.

I thank my friends, Victoria Passov-Mazo and Eli Passov for their insights into machine learning, continuous support and encouragement through the hard times.

Finally, I would like to thank my husband, my family and friends for their continuous support throughout my research.

Contents

Chapter 1

Introduction

The acquisition of the morphophonological part of the grammar is an unsupervised learning process supported by positive evidence alone. A child acquires the URs and constraint rankings using distributional cues only (Calamaro and Jarosz, 2015). There is no direct feedback during language acquisition and no direct information about paradigms. A continuous stream of speech does not contain reliable pauses or any other language-independent cues, except for the acoustic and statistical cues, to aid with speech segmentation and acquisition of applicable constraints. These distributional cues include transitional probabilities and statistical information related to sequences of linguistic units. The acquisition of phonotactic and morphophonemic constraints, as well as the acquisition of URs, is interdependent and must occur simultaneously. To model these acquisition processes, various approaches and models have been proposed in an attempt to simulate possible methods of search for the optimal hypothesis in an infinite UG space

and induction of the correct governing laws for a given language.

A morphophonological grammar consists of a lexicon with URs, their ordering, and a constraint hierarchy governing the input-output relations between the URs and the surface forms. The morphological part of the grammar is represented by relations between affixes and their order, and the phonological part – by relations of these affixes with the phonotactic constraints and their ranking. The task of an OT learner is to induce a hypothesis of a grammar, which, given inputs, can generate all outputs of the language, and to present the hypothesis applicable to this language in a compact and precise manner. A central challenge for learning algorithms is the subset problem (Angluin, 1980; Baker, 1979). This problem occurs when, besides all the forms in a given language, a learner's grammar produces additional forms that are not present in the language – in other words, during the search for a hypothesis, the learner will end up with an overgeneralizing grammar. If a learner receives only positive evidence, it gets no explicit instruction that these additional non-existing forms are non-grammatical, which can prevent the learner to make further corrections to its current hypothesis and result in the target language being a subset of the learner's current hypothesis. Therefore, a method to restrain the learning procedure would be to consider more restrictive hypotheses before considering their supersets. On the other hand, too much restriction during the learning process may result in not generalizing at all and overfitting the data.

The Sound Pattern of English by Chomsky and Halle (1968) provided an elaborate description of phonological processes, phonology acquisition theory, as well as an evaluation criterion for learning grammars – while searching the hypothesis

space for an optimal grammar, shorter grammars must be favored over long ones: for a grammar *G*, which can parse the data, the value of *G* is $\frac{1}{|G|}$, i.e., the inverse of the length of *G*. Their work discussed the criterion of "simplicity" (introduced in Chomsky (1951), Halle (1962)) or the "economy criterion", according to which the optimal grammar needs to describe the data easily and without unnecessary complexities while analyzing the data.

Although this evaluation criterion was appealing as a general basis for comparing UG theories, the preference for simple grammars would lead to substantial deficiencies when evaluating hypotheses. For example, when choosing between a grammar with restricted optionality and its simpler superset, the evaluation metric would prefer the superset grammar. The general challenge of restricted optionality was pointed out by Braine (1971), and Baker (1979) showed detailed case studies in this respect within syntax. Within phonology, Dell (1981) demonstrated that when restricted optionality is present in phonological grammars, the evaluation metric would yield incorrect hypotheses. One of Dell's examples was the optional *l*-deletion in French: a word-final liquid is optionally dropped before a pause or a consonant if it is preceded by an obstruent. This places a restriction on the phonological grammar, e.g. *quelle table?* 'which table?' can be pronounced as [kɛltabl] and [kɛltab], while *parle* 'speak' will always be pronounced [parl] and never *[par], since, in the latter, the final *l* is not immediately preceded by an obstruent. Therefore, the environment for *l*-deletion is restricted. Chomsky and Halle's simplicity criterion would favor an overgenerating grammar, which would allow *l* to be deleted after any consonant, and dismiss the more complex grammar

with a restrictive environment for optional deletion, leading to the subset problem. This, in turn, illustrates the evidence for a more general problem – focusing on the economy criterion alone and not restricting the hypothesis in any way leads to overgeneralizing grammars, as discussed in Rasin et al. (2017).

The evaluation metric for hypothesis search proposed in SPE inspired subsequent work, but it has not been actually used in a learning algorithm. We will set the discussion of rule-based phonology learning challenges aside, since they are not the purpose of this paper, and focus on learning in OT.

Prince and Smolensky's Optimality Theory (1993) gave rise to new ideas in the phonology learning field and resulted in a number of learning algorithms as well. Unlike the explicit context-sensitive rewrite rules of SPE, OT is based on a hierarchy of universal constraints and their ranking within it, which affects the relation between the surface forms and URs. In OT, the optimal output among the infinite range of possible outputs is that which optimally satisfies the constraint ranking. Thus, the steps of analyzing an unknown language in terms of morphophonology would be the induction of the lexicon consisting of morphemes, their order, as well as phonological constraints and their ranking, obtained simultaneously, while choosing the right UR after the evaluation of the competing hypotheses. [1]

The proposed OT learners followed, *inter alia*, the Richness of the Base principle (ROTB, Prince and Smolensky, 1993; Smolensky, 1996) and Lexicon Optimization principle (LO, Prince and Smolensky, 1993; Inkelas, 1995):

[1] Although all theories agree that the lexicon and the constraint rankings are acquired, the question whether the constraints themselves are acquired is being disputed.

1. Under ROTB principle, which holds that all inputs are possible in all languages, distributional and inventory regularities follow from the way the universal input set is mapped onto an output set by the grammar, a language-particular ranking of the constraints. (Prince and Smolensky, 1993:209)

2. According to Lexicon Optimization principle, when output candidates are penalized under language-particular constraint rankings, only URs for optimal candidates under these constraint rankings will be stored. Lexicon Optimization will always store the most harmonic candidate, that is, the chosen UR will be the one that maps onto the surface form with the minimal number of constraint violations.

For example, as Booij (2011) points out in his work on morpheme structure constraints, while there is no input constraint which prohibits the morpheme **bnik* in English, the Markedness and Faithfulness constraints imposed on the output will yield a different morpheme as the optimal surface form, which is non-faithful to input, e.g., *blik*, with Markedness constraints applied. Since surface forms like **bnik* do not exist in English, the UR *bnik* will not be stored in accordance with LO. Following the ROTB principle, the learning is based on morphophonological surface form alternations. Within this framework, extracting the information about alternations is based on *paradigmatically-related* surface form pairs, which helps learn the properties of URs − given a surface form pair in Russian [*gorot*] 'city.sg' and [*goroda*] 'city.pl', a paradigm-based learner may conclude that the UR in this case is */gorod/*, which is not identical to the surface form [*gorot*]. When there

are no alternations, the surface forms are faithful to URs and follow the Lexicon Optimization principle.

The models based on ROTB and Lexicon Optimization mostly targeted specific learning problems in theory, rather than providing an all-encompassing evaluation metric for the components of UG, and were not able to address the covert interaction of phonological structures in the absence of alternations. In particular, learning from alternations posits an issue in regards to morphophonology − paradigm-based phonotactic learners use identity maps from the lexicon URs to the surface forms, however, using identity maps for morphophonemic alternations would not suffice. Based on the evidence from coalescence in Sanskrit, Rotuman and Choctaw, as well as opacity and allophony in Japanese, McCarthy (2005) discussed the issue of non-alternating forms derived from unfaithful maps and demonstrated that in the absence of relevant morphophonemic alternations a learner would generalize the unfaithful map across the entire language. Therefore, McCarthy proposed the Free-Ride principle for morphophonemic learning based on non-alternating forms, which prevents the generalization by dividing the learning process into stages in order to find the more restrictive grammar before proceeding with the overgeneralizing hypothesis. However, there are empirical challenges to the Free-Ride principle in the literature, as in Nevins and Vaux (2007) (see also Krämer (2012) for discussion), which are briefly discussed in Chapter 4.

As exemplified by stress-epenthesis interaction in Yimas, Mohawk, and Selayarese by Alderete and Tesar (2002), the paradigm-based approach leads the

learner to commit to superset grammars when presented with the data that contains no alternations. In other words, when there is an identity map between the surface form and the lexicon UR, the learner will overgeneralize. Alderete and Tesar pointed out that in order for a learner to acquire non-alternating URs which are distinct from their surface forms, the models based on constraint re-ranking must be modified, and argued that learning must occur even in the absence of alternations, which requires acquisition of URs not identical to surface forms. If Alderete and Tesar are right about their claim regarding the learning in the absence of alternations, then those are further cases that Free-Ride cannot account for.

The common feature of the paradigm-based learner approaches proposed by Tesar (2006, 2009, 2014), Apoussidou (2007), Merchant (2008), and Akers (2012) was that they allowed for data acquisition by relying on grammatical alternations in order to support the learning of UR properties within discrete, restrictive OT grammars. However, these models did not present the solution for the acquisition of non-alternating URs with non-identical surface mappings. An explicit paradigm-based learner utilizing proposed modifications to resolve this issue is still a task for the future, and the challenge of learning unfaithful maps from non-alternating forms still remains.

In contrast to paradigm-based learners, which offered no generalization regarding non-alternating URs, a number of probabilistic OT models have been proposed, such as Maximum Likelihood Learning of Lexicons and Grammars (Jarosz, 2006), which combined the advantages of the paradigm-based learner

with the stochastic grammar approach and addressed morpheme-specific lexicon learning, and Lexicon Entropy Learner (Riggle, 2006a), which suggested an approach to the economy measure. However, the economy-restrictiveness balance of the proposed learners would shift towards the former or the latter, which would again result in either overgeneralization or overfitting, which is discussed in detail in Chapter 4.

The phonological OT learner proposed by Rasin and Katzir (2016) uses the Minimum Description Length criterion (further discussed in Chapter 2) as a model for language acquisition, while preserving the economy-restrictiveness balance. This was the first learner which managed to fully induce the grammar and to succeed in learning optionality and morphophonological alternations, as well as non-alternating URs distributionally, from unanalyzed surface forms alone. To further demonstrate the effectiveness of the MDL approach, Rasin et al. (2017) have developed a fully distributional MDL-based SPE learner, which manages to jointly acquire phonology and morphology and induce morphological voicing assimilation, rule interaction and opacity.

The present work proposes a further extension of the MDL-based phonological learner into a morphological learner within the framework of OT, as the first attempt to demonstrate that morphophonological OT learning can be achieved through the acquisition of constraint rankings, while jointly learning phonology and morphology. Our learner is fully distributional – it starts with a corpus of unanalyzed surface forms without any cues, feedback or indication about any paradigmatic relations between these forms. It is presented with sets of surface

forms and attempts to induce the URs and the optimal Markedness and Faithfulness constraint hierarchy, under which the optimal UR candidates can be generated. We will model the learning of voicing assimilation based on plural English forms (the assimilation of morpheme /z/ to the voice feature of the final segment of the preceding morpheme), morphophonology acquisition with inter-morphemic and an inter-phonemic epenthesis using the toy *ab-nese* language from Rasin and Katzir (2016), and conduct a preliminary investigation with vowel harmony to illustrate a general idea of the process where a suffix vowel mirrors the vowel features of the preceding stem morpheme.

We start with Chapter 2, where we outline the framework of the MDL-based learning, the hypothesis representation, the details of the learner itself, focusing on the description of various automata utilized to represent grammar components, following Riggle's weighted Finite State model for constraints and using the Hidden Markov Model (HMM) along with a nondeterministic finite automaton (NFA) for lexicon and parsing respectively, and the Simulated Annealing algorithm for searching hypothesis space. In Chapter 3 we will present the results of simulations with joint phonology and morphology learning. Chapter 4 will review the previous OT learning models, discussing the paradigm-based learners proposed by Prince and Smolensky (1993), Smolensky (1996) and Tesar and Prince (2003), Maximum-Likelihood Learning of Lexicons and Grammars (Jarosz, 2006) and Lexical-Entropy Learning (Riggle, 2004), and evaluating them in terms of economy-restrictiveness balance. Chapter 5 will conclude.

Chapter 2

Present work

2.1 MDL-based learning

According to the principle of Minimum Description Length (MDL), the best hypothesis to describe the data is the one which compresses the data the most. That is, the best model minimizes the overall description of data measured in bits, which is represented by encoding lengths of the data given the model ($D|M$) and the prior of the model (M). The MDL principle was first formulated by Solomonoff (1964), and later independently rediscovered by Kolmogorov (1965) and Chaitin (1966) as an idea to view hypotheses as programs that output the data, and to evaluate these hypotheses in terms of their lengths. *Kolmogorov complexity* is the length of the shortest program, which, given a string, produces it as an output, and then halts. Kolmogorov complexity is not computable[1], but it serves as

[1] See Li and Vitányi (2008) for a detailed discussion of Kolmogorov complexity

an important tool for evaluating learnability. In order to ensure computability, the hypothesis space must be restricted, which is done within the framework of MDL. MDL and related Bayesian approaches have been used to address the learning of various aspects of linguistic knowledge by Berwick (1982), Rissanen and Ristad (1994), Stolcke (1994), Brent and Cartwright (1996), Grünwald (1996), de Marcken (1996), Clark (2001), Goldsmith (2001, 2010), Dowman (2007), Chater and Vitányi (2007), Hsu and Chater (2010), and Hsu et al. (2011), among others.

If we apply the MDL principle to grammar learning, the prior of the model would be the description of the grammar itself (*G*), which consists of the lexicon and grammar rules. Then, the likelihood of the data is how well this grammar describes the given data ($D|G$). Given these two components, we optimize over their values simultaneously, i.e. $|G|+ |D{:}G|$. We want *G* to be as *compact* as possible, and $D|G$ as *restrictive* as possible − in other words, the grammar should be able to generate all forms of the language and describe the data easily in the least complex way.

Rasin and Katzir (2016) proposed the MDL-based learner within the framework of OT, which searches for optimal hypothesis by maximizing the *economy* (compactness) and *restrictiveness* of the grammar in terms of encoding lengths of the grammar's components, measured in bits[2]. The encoding length of *G* represents the economy (and corresponds to the "simplicity" or "economy criterion" in Chomsky and Halle (1968) − the ability of the grammar to describe the data easily, avoiding unnecessary complexity in its analysis). The encoding length of $D|G$

[2]For an argument for MDL as a null hypothesis for acquisition see Katzir (2014).

represents the restrictiveness – a grammar, which requires fewer bits to encode the data, will consider the data typical and deviations from it as special cases, and will generate only the forms, which have been observed. The overall description of an OT grammar is, therefore, $|G|+ |D{:}G|$, measured in bits.

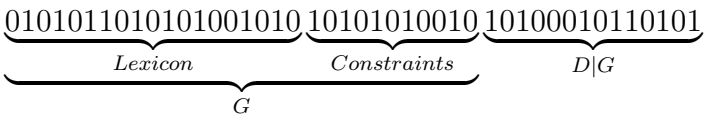


Figure 2.1: Schematic view of Solomonoff's evaluation metric as applied to OT. The grammar G consists of both Lexicon and Constraints. The data D are represented not directly but as encoded by G. The overall description of the data is the combination of G and $D|G$.
Source: Rasin and Katzir (2016).

In their attempts to escape the subset problem, previously proposed probabilistic OT models (Jarosz, 2006; Riggle, 2004), which are further discussed in Chapter 4, used either restrictiveness or economy criteria (but not together and/or not equally considered). The issues of those models pointed to the fact that in order for the hypothesis to be well-formed and to allow for less errors, the economy-restrictiveness balance must be maintained, and both criteria must be maximized together. A learner without the restrictiveness-economy balance faces the risk of overgeneralizing or not generalizing at all (as discussed in Chapter 4). This balance is achieved by minimizing the sum of the grammar's encoding length and the encoding length of the data given the grammar:

$$\arg\min_G\{|G| + |D{:}G|\}$$

The model based on the MDL evaluation metric proposed by Rasin and Katzir

(2016) demonstrated unsupervised induction of a lexicon and a phonological grammar within the framework of OT. The learner was tested with corpora containing surface forms modeled after the English aspiration, e.g. $[k]$ in $k^h æt$, the French optionality of $[l]$ in *table* 'table' versus this segment being obligatory in *parle* 'speak' (the example from Dell, 1981), the Hebrew voicing assimilation, e.g. *katav* $\rightarrow$ *kataft* 'he wrote' $\rightarrow$ 'you (*2fs*) wrote', and grammar induction without relying on alternations. Presented with unanalyzed surface forms, the learner succeeded in arriving at correct hypotheses.

The MDL evaluation metric was further tested by Berger (2018) and (Rasin et al., 2017) within the framework of SPE, resulting in the first fully distributional morphophonological learner based on context-sensitive rewrite rules, which has succeeded in inducing phonological rules of voicing assimilation and optionality, and demonstrated joint learning of morphology and phonology, as well as rule ordering and opacity.

This work proposes a morphophonological learner within the framework of OT, by extending the phonological learner of Rasin and Katzir. In order to demonstrate the joint morphology and phonology learning, the learner will be presented with three datasets represented by corpora containing morphophonological patterns without any indication of morpheme boundaries, and with constraint sets initially containing either a single FAITH constraint, or a constraint hierarchy in a reversed order.

(1) A **voicing assimilation** example – plural voiced consonant in the suffix devoices after a voiceless obstruent. The learner is presented with the cor-

pus modeled after the English voicing assimilation in plural forms, where the UR suffix morpheme /z/ devoices after a voiceless obstruent in the stem, e.g. /katz/→ [kats], /dogz/→ [dogz].

Our corpus consisted of the following surface forms:

[‘dag’, ‘kat’, ‘dot’, ‘kod’, ‘gas’, ‘toz’, ‘ata’, ‘aso’,

‘dagdod’, ‘daggos’, ‘dagzook’,

‘kattod’, ‘katkos’, ‘katsook’,

‘dottod’, ‘dotkos’, ‘dotsook’,

‘koddod’, ‘kodgos’, ‘kodzook’,

‘gastod’, ‘gaskos’, ‘gassook’,

‘tozdod’, ‘tozgos’, ‘tozzook’,

‘atadod’, ‘atagos’, ‘atazook’,

‘asodod’, ‘asogos’, ‘asozook’]

The first line of the corpus above contains 3-segment stems without suffixes. The rest of the corpus contains these stems concatenated with 3 UR suffixes, ‘zook’, ‘gos’, ‘dod’, the initial consonants of which undergo devoicing due to the assimilation with the last consonant in some of the stems. The goal of the learner is to learn suffix and stem URs, and to induce the voicing assimilation in the surface forms, which is dependent on the presence of the last voiceless obstruent consonant in the stem. In other words, presented with the corpus above, the learner is supposed to infer the distinction between the stem and suffix morphemes, and to arrive at the correct constraint hierarchy, under which morphemes such as ‘tod’,

'kos', 'sook' are the result of assimilation-enforcing constraint ranking, and the correct parses from the UR to the surface form would be /katzook/ $\rightarrow$ [katsook], /dagzook/ $\rightarrow$ [dagzook], etc.

The learner starts with the voicing assimilation constraint ranking in reverse and a lexicon identical to the data above.

Initial hypothesis:

$$G_{initial} = \begin{cases} \textsc{Lex}: & dagzook, katsook, dagdod, dottod, tozgos, gaskos, \\ & kat, toz \ldots \\ \textsc{Con}: & \textsc{Ident}([-velar]) \gg \textsc{Ident}([-strident]) \gg \\ & \gg \textsc{Ident}([+velar]) \gg \textsc{Ident}([+strident]) \gg \\ & \gg \textsc{Ident}([+cons][-voice]) \gg \textsc{Ident}([+cons]) \gg \\ & \gg \textsc{Faith} \gg * \begin{bmatrix} +cons \\ -voice \end{bmatrix} \begin{bmatrix} +cons \\ +voice \end{bmatrix} \gg \\ & \gg \textsc{Dep}([-cons]) \gg \textsc{Max}([+cons]) \end{cases}$$

(2) A **complex morphophonology** example – inter-morphemic and inter-phonemic epenthesis.

In the initial hypothesis, the constraint set contains only FAITH, and the lexicon is identical to the data with words generated by appending the prefix *aab* to various stems. The goal of the learner is to induce the correct constraint hierarchy, the prefix morpheme and the epenthesis of *a* between

the final *b* in the prefix and the initial *b* in the stem, as well as the epenthesis of *a* between the *bb* sequences in the stems, e.g. UR $/aabbbab/ \rightarrow$ Surface form $[aabAbAbab]$ (the epenthetic *a*'s are capitalized).

Initial hypothesis:

$$G_{initial} = \begin{cases} \text{LEX:} & ab, ba, baba, aabab, aababa, aababababab \ldots \\ \text{CON:} & \text{FAITH} \end{cases}$$

(3) A **vowel harmony** example − the vowel in the suffix corresponds in its feature to the vowel in the stem. In this simulation, the learner will be presented with the following corpus:

['unu', 'uku', 'nunu', 'kunu', 'nuku', 'kuku',

'ini', 'iki', 'nini', 'kini', 'niki', 'kiki',

'unukun', 'ukukun', 'nunukun', 'kunukun', 'nukukun', 'kukukun',

'inikin', 'ikikin', 'ninikin', 'kinikin', 'nikikin', 'kikikin']

The corpus consists of 12 stems, half of which contain the [+back] vowel 'u' and the other half contains the [−back] vowel 'i'. These stem morphemes are then concatenated with the UR suffix /kun/, which changes to [kin] under the vowel harmony enforcing constraint ranking. The goal of the learner is to induce the UR morphemes of the lexicon, and the constraint ranking, under which the [± back] feature of the stem vowel spreads onto the target vowel in the suffix, resulting in parses such as /ukukun/ → [ukukun], /inikun/ → [inikin]. The constraint ranking pre-

sented to the learner at the initial step is the reversed hierarchy of the vowel harmony enforcing constraint set.

Initial hypothesis:

$$G_{initial} = \begin{cases} \textsc{Lex}: & unu, uku, ini, iki, nini, unukun, ukukun \\ & inikin, ikikin, ninikin \ldots \\ \textsc{Con}: & \textsc{Faith}([-velar]) \gg \textsc{Ident}([-cons]) \gg \\ & \gg \textsc{Ident}([+cons]) \gg \textsc{Dep}([+cons]) \gg \\ & \gg \textsc{Dep}([-cons]) \gg \textsc{Max}([+cons]) \gg \\ & \gg \textsc{Max}([-cons]) \gg \\ & \gg * \begin{bmatrix} -cons \\ +back \end{bmatrix} [+cons] \gg \begin{bmatrix} -cons \\ -back \end{bmatrix} \gg \\ & \gg * \begin{bmatrix} -cons \\ -back \end{bmatrix} [+cons] \gg \begin{bmatrix} -cons \\ +back \end{bmatrix} \gg \\ & \gg \textsc{Dep}([-cons]) \gg \textsc{Max}([+cons]) \end{cases}$$

Before we proceed with describing each simulation in detail and reporting the results in Chapter 3, it is important to provide the explanation about the building blocks of our model.

2.2 Hypothesis representation

In order to make the model computationally viable, the representations of the grammar components and the data follow the finite-state OT framework implementation, which involves finite-state automata for the purpose of encoding input/output mappings, affixes and stems ordering, and parsing facilitation via removal of redundant candidates. To calculate the encoding lengths of data and grammar within the MDL framework, data strings and automata must be encoded as binary strings.

2.2.1 Automata

The goal of this learner is to derive optimal UR candidates in terms of code length given the grammar and the surface form data. To reach this goal by generating all possible derivations from a grammar would not be achievable even with simple grammars, therefore, the lexicon and constraints are presented as finite state automata (FSA). Our constraint hierarchy representation is based on the weighted finite-state OT model developed by Riggle (2004), where each constraint is represented by a finite state transducer (FST), and obtaining the EVAL constraint set follows Riggle's intersection rules for constraint FSTs. The lexicon is represented by the Hidden Markov Model (HMM), which undergoes further transformations into a non-deterministic automaton (NFA) for the purposes of generation and parsing. When evaluating $D|G$ during output generation, each input word of the lexicon is represented as a finite state input acceptor, which is intersected with the

constraint set FST to generate the optimal output given the constraints. The HMM and the constraint set FSTs represent the grammar components to be mutated during the search for the best hypothesis, and, together with the parsed data given the grammar, evaluated in terms of encoding length.

2.2.1.1 Lexicon HMM

Since the goal of this research is to model the acquisition of morphophonological processes, the lexicon must be represented in a way that allows selective morpheme combinations and specific ordering of the morphemes. The Hidden Markov Model allows us to store the morphemes in the HMM emission table, and morpheme combinations can be defined by state transitions. In the example in Figure 2.2, the lexicon has the stems /dog/ and /kat/, and the optional suffix /z/.

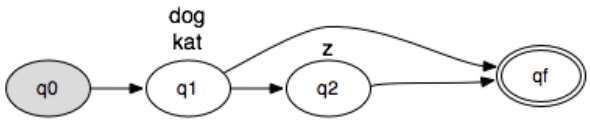


Figure 2.2: Plural English lexicon represented by an HMM

The lexicon segments are represented as feature bundles, contained within the feature table (as in Figure 2.11), which is provided to the learner before a simulation begins:

	cons	*voice*	*velar*	*cont*	*low*	*strident*
a	−	+	−	+	+	−
d	+	+	−	−	−	−
g	+	+	+	−	−	−
k	+	−	+	−	−	−
o	−	+	−	+	−	−
s	+	−	−	+	−	+
t	+	−	−	−	−	−
z	+	+	−	+	−	+

Figure 2.3: Feature table

The HMM is implemented by creating the list of states, the emission dictionary, and the transition dictionary. The list of states will contain the initial, inner, and final states. The emission dictionary is constructed by taking each word from the data and assigning it as an emission of an inner state. The transition dictionary will contain the transitions between the states.

In order to apply phonotactic and morphophonological rules to the emissions of the HMM, it needs to be converted into a NFA, so that the HMM emissions are broken down into segments with their corresponding features from the feature table. The initial HMM is as shown in Figure 2.2. The NFA is constructed using FADo (an automata manipulation library for Python[3]) as follows:

1. Creating an empty instance of a FADo NFA, assigning the initial state (*q0)* and the final state (qf) of the HMM to be the initial and final states of the NFA and appending them to the list of NFA states to be populated when iterating over the HMM.

[3]FADo documentation can be found at http://fado.dcc.fc.up.pt/

2. Creating a mapping dictionary of states and their transitions represented by tuples, e.g. $\{q0 : \{StateTuple(q_{start}, q_{end})\} \dots\}$ and iterating over the inner states of the HMM to populate the dictionary with (q_{start}, q_{end}) tuples. Extending the list of NFA states after the iterations with the start and end states.

3. Iterating over the HMM emissions and creating a list of emissions-by-state, containing the state, emission index, and segment index. (For HMM with emission dictionary $\{q1 : [\text{“}dog\text{”}, \text{“}kat\text{”}], q2 : [\text{“}z\text{”}]\}$, the emission segment *k* would be represented as $[q1, 1, 0]$. After the iteration over emissions, the list of NFA states is further extended with the emissions-by-state list values.

4. Iterating over the list of NFA states, constructing a dictionary with the states as keys and assigning numeric state indices as their values. Setting the initial and final NFA states to those from the $\{$*state : state index*$\}$ dictionary.

5. Iterating over HMM transitions and emissions to create transition arcs for the NFA, based on the NFA state list and HMM emission dictionary.

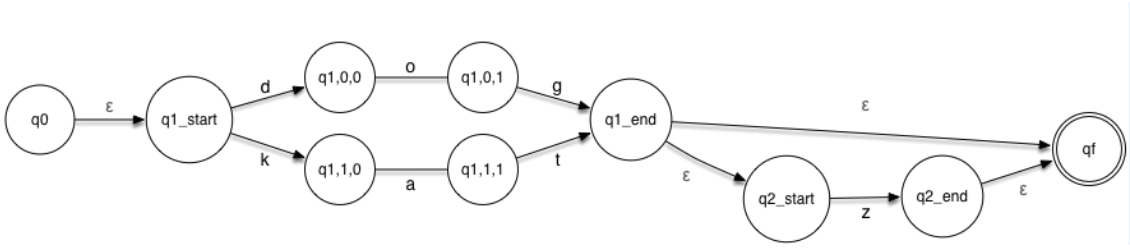


Figure 2.4: HMM converted into NFA

During the learning itself, the learner is presented with unanalyzed surface

forms consisting out of stems and stem-suffix combinations in order to induce the optimal hypothesis. At the initial step, the words in lexicon are equal to the data surface forms, and neither defined ordering nor any information regarding morphemic segmentation into stems and affixes is provided to the learner. The learner's task is, therefore, to induce the correct morphophonological constraint hierarchy and the ordering of morphemes, and to output the constraint ranking, the lexicon in the form of HMM, and the encoding length for the overall description of the grammar. For example, if the goal of the learner to induce the morphophonological laws for $kat \rightarrow kats, dog \rightarrow dogz$, the initial HMM will be as follows:

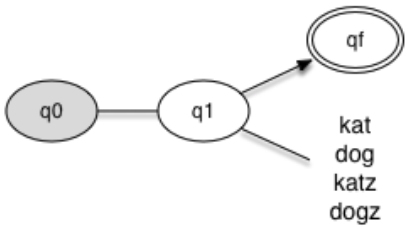


Figure 2.5: Initial "naive" HMM created from list of strings [*kat, dog, katz, dogz*]

As shown in Figure 2.5, the initial HMM has only one inner state and there is no defined ordering or separation of the morphemes into stems and affixes.

2.2.1.2 Constraint set FST

Each constraint is represented as a weighted finite-state transducer (wFST) − an automaton, where each arc is a 5-tuple containing the **origin** of the arc, an **input**, an **output**, a binary **cost** vector to indicate whether a segment violates a constraint during i/o mapping, and the **terminus** of the arc:

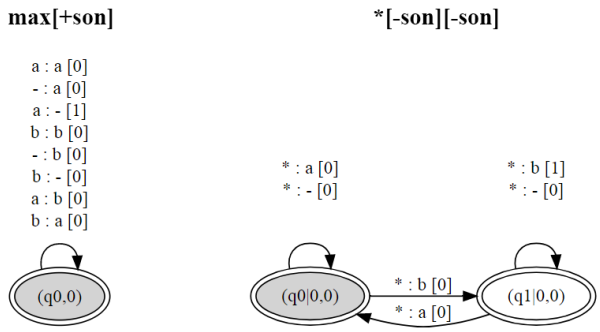


Figure 2.6: MAX[+son] transducer and Phonotactic [-son][-son] transducer

Figure 2.6 illustrates the Faithfulness constraint MAX and the Markedness phonotactic constraint transducers with the alphabet consisting of two segments, {a, b}. MAX[+son] penalizes the deletion of sonorants ('a'), and *[-son][-son] penalizes non-sonorant 'bb' sequences. The inputs and outputs on the arcs are separated by a colon, and the cost vectors are contained in the square brackets. Violations are represented by non-zero weights in the cost vectors. For example, the cost vector of the i/o sequence *a:-* in MAX[+son] is [1], which means that upon receiving 'a' as the input and a null segment '-' as the output (i.e. the input segment gets deleted), the constraint transducer for MAX[+son] marks it as a violation by assigning a weight of 1 to its cost vector. The rest of the arcs get a zero cost vector, since no violations occurred.

While Faithfulness constraints, like MAX, DEP, IDENT produce single-state transducers, Markedness constraint transducers are more complex and require multiple states to define the environments where violations can occur. In Figure 2.6, the input slots of the phonotactic constraint *[-son][-son] are filled with wildcard segments (*), indicating that any input is allowed, since phonotactic con-

straints are focused on the output properties. As we can see in the phonotactic constraint transducer above, no matter what the input can be, a sequence of two *b*'s will result in a violation and a non-zero cost vector.

The constraint hierarchy relevant to the grammar we are working with is represented by a constraint set transducer, or EVAL transducer. This transducer is created by obtaining the Cartesian product of the states of each constraint transducer, and by unifying the arc inputs when they hold the same segment or when one of the arcs has a wildcard segment input, as follows:

(4)

$$x \cup y = \begin{cases} \text{x} & \text{if} \quad \text{x = y} \\ \text{y} & \text{if} \quad \text{x = *} \\ \text{x} & \text{if} \quad \text{y = *} \\ 0 & \quad \text{otherwise} \end{cases}$$

The cost vectors of each arc are concatenated in order of intersection.

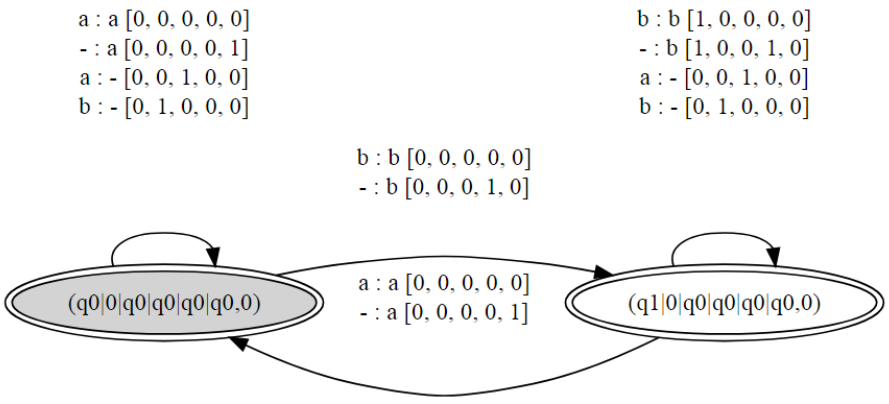


Figure 2.7: EVAL transducer resulting from the intersection of *[-son][-son] ≫ MAX[-son] ≫ MAX[+son] ≫ DEP[-son] ≫ DEP[+son]

The optimal candidate is chosen on the basis of *harmony*. A candidate is harmonic if it satisfies the highest-ranking constraint the best. Given two cost vectors, *v* and *w*, the former is more harmonic than the latter, if for every constraint C_j for which *w* has fewer violations than *v* there is some constraint C_i ranked above C_j for which *v* has fewer violations than *w*. For example, the vector [0,0,0,0,1] in Figure 2.7 is more harmonic than [0,0,1,0,0]. Since this definition of harmony guarantees that no two distinct vectors can be equally harmonic, there will always be a single unique *most harmonic vector* in any set of cost vectors.

To perform optimization for an input word after EVAL transducer is constructed by intersecting constraint transducers (following Riggle's machine intersection procedure), the input word is represented as an input acceptor (as in Figure 2.8) to be intersected with the EVAL. In order to reduce the complexity of EVAL and word generation, the intersected transducers are further optimized. The initial intersection of transducers may produce dead states, which will not participate in optimal candidate search, that is, *unreachable* and *impasse* states. These states are recursively removed from the initial EVAL transducer, and the same procedure is repeated after EVAL is intersected with the word transducer.

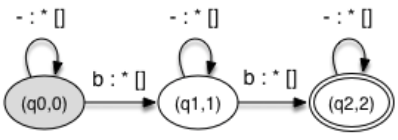


Figure 2.8: /bb/ input acceptor

In addition to dead state removal, it is important to ensure that the final trans-

ducer will contain all and only the most harmonic paths by removing the *suboptimal paths* from the initial transducer, so that the final transducer can generate all and only the optimal candidates. This is done by maintaining the minimal path cost from the initial state to each transducer state, and then removing the arcs that are not involved in any optimal path. The outputs are then populated with a set of strings created by the outputs of the optimal paths in the initial transducer.

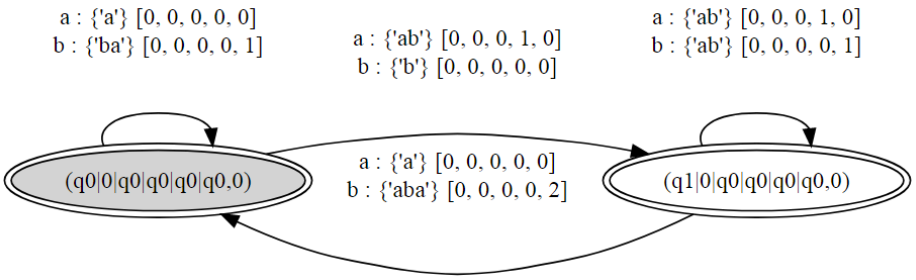


Figure 2.9: EVAL transducer after dead states and suboptimal paths removal

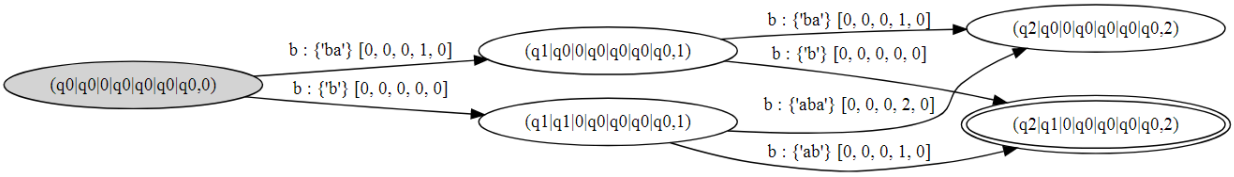


Figure 2.10: Final optimized EVAL transducer intersected with /bb/ input acceptor

2.2.2 Encoding length

Based on the MDL principle and the evaluation metric presented by Rasin and Katzir (2016), and, as shown in Figure 2.1, the total encoding length of the grammar would constitute a sum of its components' encoding lengths:

(5) $|code(G)| = |code(\text{LEX})| + |code(\text{CON})|$

2.2.2.1 Lexicon encoding length

To provide the initial explanation of the encoding technique, we will start with a simple example of encoding a sequence of lexicon words (HMM emissions). As mentioned in the section describing the HMM, which represents the lexicon, each segment of HMM emissions is represented as a feature bundle in the feature table:

	a	b	s
$cons$	$-$	$+$	$+$
$cont$	$+$	$-$	$+$

Figure 2.11: A feature table for the alphabet *a, b, s* with *consonantal* and *continuant* binary features

Given the list of emissions in (6a), and using a delimiter ($\#_w$) to mark the end of each word, as well as the end of the entire sequence of emissions, the string representation of this list (based on the feature table in Figure 2.11) will be as shown in (6b). Each string representation symbol in (6b) is then substituted with a two-digit binary code, i.e. 00 for $+$, 01 for $-$, and 10 for #, and its encoding is shown in (6b). The size of the emission sequence will be the length of the string in (6c).

(6) a. $[asa, ba, bsab]$

b. $-++++-+\#_w+--+\#_w+-+++-+++-\#_w\#_w$

c. $01000000010010000101001000010000010000011010$

In general, if the length of the lexicon's alphabet is *n*, each individual segment would need to be encoded with $\lceil \lg n \rceil$ bits. For example, the alphabet of

our {katz, dogz} voicing assimilation example consists of 8 segments, and the number of bits to encode it would be $\lceil \lg(8) \rceil = 3$ bits for each segment, times the number of characters in the sequence, including commas. Although the IPA alphabet has 107 letters and 31 diacritics, we are not taking into account all of its segments in order to evaluate the number of bits to encode our sequence, since no segments other than a, d, g, k, o, s, t, z are present in it, and therefore the description of the corresponding grammar would be shorter and more compact. At the same time, we would be imposing a restriction ($*\neg$) on our lexicon's alphabet: $*\neg a, d, g, k, o, s, t, z$, which would have to be added to the grammar's description, therefore slightly increasing the size of *G* (compared to a grammar without any restrictions), but, following this alphabet restriction, the savings for $D|G$ description in bits compensate this addition by requiring 3 bits instead of 8. [4]

Since the lexicon in this model is represented by the HMM, besides encoding the sequence of words (emissions), we need to encode the information about morpheme ordering, as well as the states and transitions of the HMM. Let us demonstrate this with a string representation of the plural English HMM presented above in Figure 2.2:

State	Code
q_0	$q_0 q_1 \#_S \#_w$
q_1	$q_1 q_2 q_f \#_S$dog$\#_w$kat$\#_w \#_w$
q_2	$q_2 q_f \#_S$z$\#_w \#_w$

Figure 2.12: String representations of HMM states

[4]Naturally, there are other ways to encode different alphabets, which will affect the value of $|G|$ accordingly.

In Figure 2.12, each state is encoded as a sequence of its emissions and transitions. Each state transition ends with a delimiter $\#_S$ to indicate the end of state transition, and each emission for that state ends with the $\#_w$ delimiter to indicate the end of word and the end of word sequence.

The string representation for the entire HMM is as follows:

$$q_0 q_1 \#_S \#_w \#_w q_1 q_2 q_f \#_S dog \#_w kat \#_w \#_w q_2 q_f \#_S z \#_w \#_w$$

Figure 2.13: String representation of an HMM

Symbol	Code
q_0	001
q_1	010
q_2	011
q_f	100

Symbol	Code
$\#_S$	000

Symbol	Code
k	0001
a	0001
t	0010
...	...

Symbol	Code
$\#_w$	0000

Figure 2.14: Binary code assigned to each HMM symbol.

Each state of the HMM can be described in terms of its symbol, its transitions, its emissions, and number of segments in its emissions. The encoding length of state symbols for the HMM ($|code(Q)|$) would be $\lceil \log_2(n_q + 1) \rceil$, where n_q is the total number of states and 1 is added for the state transition delimiter $\#_S$. In our example, there are 4 states, so their encoding length would be $\lceil \log_2(4 + 1) \rceil = 3$. The encoding length of the emissions ($|code(e)|$) would be $\lceil \log_2(n_s + 1) \rceil$, where n_s stands for number of segments of the alphabet the HMM is using, and 1 is added for the morpheme delimiter $\#_w$.

The encoding length of the HMM consists of its overall content usage and delimiter usage between the state transitions, morphemes, and morpheme sequences. The content usage, $|code(content)|$ is calculated as follows:

(7)

$$|code(content)| = \sum_{t \in Q} |n_t| \cdot |code(Q)| + \sum_{s \in Q} |n_s| \cdot |code(e)|$$

where Q stands for the set of HMM states, n_t stands for the number of transitions from a state (the transitions are calculated including the origin state, e.g. for the state q_1 in Figure 2.12, which has transitions to 2 states, $n_t = 2+1$), and n_s stands for the number of segments in the state's emissions. The delimiter usage depends on the number of states and emissions of the HMM, and is calculated as follows:

(8)

$$|code(\#)| = \sum_{q \in HMM} |n_q| \cdot |code(e)| + \sum_{q \in HMM} |n_q| \cdot |code(Q)| + \sum_{e \in HMM} |n_e| \cdot |code(e)|$$

where n_e stands for number of emissions of the HMM.

The total encoding length of the HMM is:

(9)

$$|code(\textsc{Lex})| = |code(content)| + |code(\#)| + (|code(Q)| + 1)$$

The last summand is added for the generalized unary coding.

2.2.2.2 Constraint set encoding length

The constraints and their rankings are encoded in a similar fashion. The constraint hierarchy is represented as a string, and the delimiter ($\#_c$) is used to mark the end of each constraint, the end of each feature bundle (if the constraint is phonotactic),

and the end of the constraint set itself. For example, the constraint hierarchy in (10a) would be represented as the string in (10b), and then each of the symbols would be encoded according to the Figure 2.15. The letters *D*, *M*, *I*, *P* stand for DEP, MAX, IDENT, and Phonotactic constraints, while *cons* and *cont* stand for *consonantal* and *continuant* binary features.

(10) a. $\text{DEP}(-cons) \gg \text{MAX}(+cont) \gg {}^{*}[+cons] \begin{bmatrix} -cons \\ \\ +cont \end{bmatrix} \gg \text{IDENT}(-cont)$

b. $D{-}cons\#_c M{+}cont\#_c P{+}cons\#_c{-}cons{+}cont\#_c\#_c I{-}cont\#_c\#_c$

Symbol	Code
D	0000
M	0001
I	0010
P	0011

Symbol	Code
cons	0100
cont	0101

Symbol	Code
$+$	0110
$-$	0111

Symbol	Code
#	1000

Figure 2.15: Binary code assigned to each constraint set symbol.

In (10b), the initial constraint letter (*D*, *M*, *I*, *P*) indicates the constraint type and marks the beginning of the new constraint. Faithfulness constraints have one feature only and are followed by one delimiter, while Markedness constraints have bundles and need delimiters to indicate the end of each bundle, plus a delimiter for the final bundle sequence. Finally, we add a delimiter for the end of the string. After enumerating all the symbols belonging to the set (4 constraint types, 2 signs, 1 delimiter, and the number of features), we encode each symbol as $k = \lceil log_2(4 + 2 + 1 + |\text{features}|\rceil$ bits. Therefore, the total encoding length of the constraints above will be as follows:

(11) $|code(c_D)| = |code(c_M)| = |code(c_I)|$

$|code(c_P)| = k \cdot [1 + |\text{bundles in } c_P| + 2 \cdot |\text{features in } c_P|]$,

where the latter is for Markedness constraints and the former is for Faithfulness constraints. Thus, the total encoding length of CON is

(12)

$$|code(\text{CON})| = k + \sum_{c \in \text{CON}} |code(c)|$$

2.2.2.3 D|G encoding length

The second part of our hypothesis representation involves the calculation of the data given the grammar ($D|G$). Within the framework of morphophonology, surface form generation requires following specific morpheme ordering and applying morphophonological constraints. After the HMM is converted into the parsing NFA and the lexicon morphemes generated given the constraint set are concatenated and extracted, the data is being parsed. To calculate the encoding length of $D|G$, the grammar needs to describe each surface form *s* from the data presented to the learner by 1) choosing its best parse from the lexicon containing morpheme URs ($parse(s) \in$ LEX); and 2) selecting an optimal output of *s* from the set of URs, to which $parse(s)$ is mapped:

(13) $|code(s|G)| = |code(parse(s)|\text{LEX})code(s|parse(s))|$

Each choice from the lexicon (as shown in (14a)), as well as each choice from the optimal outputs (as shown in (14b)), is assigned a binary code. For example,

if we want to encode a string s_1 given the grammar, and s_1 is equal to the output $o_{1,3}$, the grammar would describe s_1 as 00010 (000 encodes the chosen UR u_1, and 10 encodes the chosen optimal output of u_1, which is $o_{1,3}$).

(14) a.

UR	Code
u_1	000
u_2	001
u_3	010
...	...

b.

u_1	
Output	Code
$o_{1,1}$	00
$o_{1,2}$	01
$o_{1,3}$	10

u_2	
Output	Code
$o_{2,1}$	-

If s_1 cannot be parsed by the grammar, its description length is set to infinity. In the case when there is more than one parse of a surface representation, which yields multiple descriptions, the shortest description will be chosen.

The total description length of $D|G$ is calculated by accessing the data presented to the learner, as well as its lexicon, and creating a data parse dictionary $\{s_i : \{u_i\}\}$. For each word in the lexicon, the grammar generates outputs given the constraint set. If the data contains the generated outputs, the parse tuple is created: (u_i, n), where n is the number of outputs u_i can generate (n indicates the number of surface forms parsed for each UR and can serve as a pointer for optionality). After that, the lexicon (HMM) is converted into the parsing NFA (an

NFA, where ε transitions are substituted by null ('-') segments to correspond to those in constraint set FST), where the length of each path is less than the length of the longest word in the data. Then, each surface form in data and its parses are evaluated in terms of their encoding lengths via the probabilistic parser by finding the most likely sequence of states and observed segments with the minimal encoding length, and then by calculating $\lceil \log_2(|\textit{outgoing states}|) \rceil$ for each NFA state, the transition of which has the observed segment.

The total data length given grammar can be formulated as:

(15)

$$|code(D|G)| = \sum_{s \in D} |code(s|G)| = \sum_{s \in D} |code(parse(s)|\textsc{Lex}) code(s|parse(s))|$$

2.3 Search

The algorithm used for searching hypotheses spaces is Simulated Annealing (SA) (Kirkpatrick et al., 1983) – a heuristic optimization technique used for a variety of problems. SA is inspired by the physical process of annealing, which is a controlled slow cooling of metal until it solidifies into a defect-free crystal state. The advantage of SA is that while searching through complicated spaces with multiple local optima, it avoids being trapped in local optima. The algorithm's random probabilistic search does not only accept changes that decrease or increase the optimization ability, but also changes, which can lead to suboptimal solutions – during its running time the probability of accepting "worse" solutions decreases.

The goal set for SA within the scope of this work is to find the global minimum in the grammar space, that is, a grammar *G* with the minimal description length. The algorithm compares a current hypothesis to its neighbors in terms of their description lengths. That is, if G' is the neighbor of the current hypothesis G, then $|G| + |D{:}G|$ is compared to $|G'| + |D{:}G'|$. If the neighbor G' is better than the initial G, the search switches to G'. If G' is worse than G, the algorithm makes a probabilistic choice whether to switch to G' or not, depending on how much worse G' is, as well as on the *temperature* parameter − the higher it is, the more likely the switch to a bad neighbor. In the beginning of the search, the temperature is set to a relatively high value. Since the probability of switching also depends on how much worse G' is, and the amount of hypotheses that are worse is unbounded in principle, there will still be many hypotheses that the search will switch to with a very low probability. While the temperature is high, in order to escape local optima, SA is allowed to accept hypotheses worse than the initial hypothesis more often. As the search progresses, the temperature gradually lowers, making the search increasingly greedy − the probability of moving to a worse hypothesis progressively changes towards 0, which allows the algorithm to focus on a search space containing hypotheses close to optimum, if the search has arrived at the neighborhood of the global optimum. Temperature lowering is performed in accordance with a cooling schedule, where the temperature at each step is multiplied by a *cooling parameter* α to yield the temperature for the next step. The search stops when the temperature reaches a defined *threshold*. The pseudocode for simulated annealing is shown in Figure 2.16 below.

$D \leftarrow$ *input string in* Σ
$G \leftarrow$ *initial_grammar(*Σ*)*
$T \leftarrow$ *initial temperature*
while $T > threshold$ **do**
  $G' \leftarrow random\_neighbor(G)$
  $\Delta \leftarrow [|G'| + |D|G'|] - [|G| + |D : G|]$
  if $\Delta < 0$ **then**
    $p \leftarrow 1$
  else
    $p \leftarrow e^{-\frac{\Delta}{T}}$
  end if
  choose $G \leftarrow G'$ with probability p
  $T \leftarrow \alpha T$
end while
return G

Figure 2.16: Simulated Annealing pseudocode.

The initial hypothesis in our case can be a grammar with the data in which no patterns have been discovered yet, with either a single faithfulness constraint FAITH, which represents an identity function between the URs and surface forms, penalizing any structural changes, or the reversed constraint hierarchy, which is given to the learner before the simulation starts (as in the simulation modeled after plural English voicing assimilation). The neighbor grammar hypothesis is generated via one of the random mutations shown in (16) for each iteration of the algorithm.

The search starts with calculating the initial hypothesis energy, when the data equals the lexicon. In order to generate the neighbor hypothesis, SA chooses one of the grammar's components to mutate – either the constraint set or the lexicon. Then, a mutation within the component is chosen at random. If the chosen ob-

ject is the lexicon, the HMM undergoes a random mutation and conversion to a new NFA, where the length of each path is set to be smaller than the length of the longest word in data. Then, the list of lexicon words is updated accordingly. For each new hypothesis during the search, the morphemes generated by the NFA given grammar are being probabilistically parsed on the basis of the data via extraction of the current lexicon URs, and checking whether a parse can be produced under the current constraint set. If the chosen object is the constraint set, the lexicon is being evaluated given the new constraint hierarchy, and new morphemes can be generated under the new ranking. Then, the neighbor hypothesis with the mutated lexicon is evaluated in terms of $|G| + |D{:}G|$ energy.

(16) **Constraint set mutations:**

a. Add constraint: A constraint with a single feature bundle is added in the constraint set.

b. Remove constraint: A constraint is removed from the constraint set.

c. Demote constraint: A constraint is demoted by one place in the constraint set.

d. Add feature bundle: A single feature bundle is added to a phonotactic constraint in the constraint set.

e. Remove feature bundle: A single feature bundle is removed from a phonotactic constraint in the constraint set.

Lexicon (HMM) mutations:

a. Add state: An empty state is added to the HMM (with no emissions

or transitions).

b. Remove state: A random state and its arcs are removed from the HMM.

c. Clone state: A random inner state is cloned together with its transitions and emissions.

d. Add emission: An emission is picked at random and added to a random inner state of the HMM.

e. Remove emission: A random emission is removed from a random state.

f. Clone emission: A random emission from the HMM emission dictionary is cloned and added to one of the inner states.

g. Advance emission: A random emission from a randomly selected start state q_{start} is added to a new state q', created between the origin and terminus of q_{start}.

h. Add transition: A new transition is added between two random states.

i. Remove transition: A transition is removed from a random state.

j. Add segment to emission: A random segment from the feature table is added as a new emission to a random state.

k. Remove segment from emission: A random segment from the feature table is removed from a random state.

l. Change segment in emission: A random segment from a random emission is replaced with another random segment.

Each mutation is chosen on the basis of uniform distribution over all possible mutations. The size of lexicon, constraint, or constraint set is not restricted.

Chapter 3

Simulations

3.1 Voicing assimilation

This simulation[1] is modeled after plural English voicing assimilation, where the UR suffix morpheme /z/ devoices after a voiceless obstruent in the stem, e.g. /katz/→ [kats], /dogz/→ [dogz]. The learner's task here is to induce morphophonological alternations, that is, to decompose the morphemes presented as unanalyzed surface forms into a lexicon of UR morphemes and to acquire both the correct morpheme ordering and the phonological constraint ranking applicable to the suffix after the end of the stem morpheme. A similar joint morphology and phonology learning model has been implemented by Rasin et al. (2017) in the SPE-based learner using the MDL principle.

The learner was presented with 32 surface forms with voicing assimilation,

[1] The code for all simulations can be found at github.com/vikkcosta/morphophonology_optimality

consisting of combinations of 8 stem and 4 suffix morphemes (including the null suffix). The initial constraint set (Figure 3.3) was presented to the learner as a reversed version of voicing assimilation constraint hierarchy, and in this case GEN was allowed to change segments besides inserting and deleting them. In the initial HMM (Figure 3.3) all the data surface forms were extracted from the list of data words and represented as emissions of its only inner state, q_1. Figures 3.1 and 3.2 show the feature table and the data morphemes respectively.

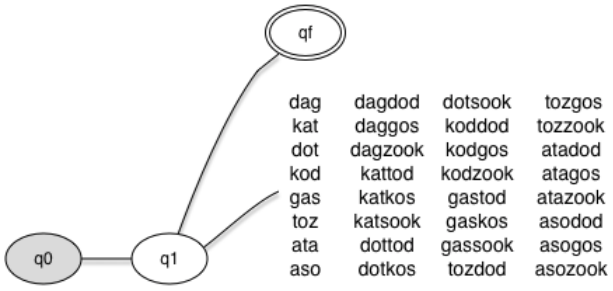
	cons	*voice*	*velar*	*cont*	*low*	*strident*
a	−	+	−	+	+	−
d	+	+	−	−	−	−
g	+	+	+	−	−	−
k	+	−	+	−	−	−
o	−	+	−	+	−	−
s	+	−	−	+	−	+
t	+	−	−	−	−	−
z	+	+	−	+	−	+

Figure 3.1: Voicing assimilation feature table

Stem	Suffix
dag	zook
kat	gos
dot	dod
kod	∅
...	

Figure 3.2: Voicing assimilation corpus stems and suffixes

Initial grammar:

$$G_{initial} = \begin{cases} \textsc{Lex}: & \\ & \text{  } \\ & \begin{array}{llll} \text{dag} & \text{dagdod} & \text{dotsook} & \text{tozgos} \\ \text{kat} & \text{daggos} & \text{koddod} & \text{tozzook} \\ \text{dot} & \text{dagzook} & \text{kodgos} & \text{atadod} \\ \text{kod} & \text{kattod} & \text{kodzook} & \text{atagos} \\ \text{gas} & \text{katkos} & \text{gastod} & \text{atazook} \\ \text{toz} & \text{katsook} & \text{gaskos} & \text{asodod} \\ \text{ata} & \text{dottod} & \text{gassook} & \text{asogos} \\ \text{aso} & \text{dotkos} & \text{tozdod} & \text{asozook} \end{array} \\ \textsc{Con}: & \textsc{Ident}([-velar]) \gg \textsc{Ident}([-strident]) \gg \\ & \gg \textsc{Ident}([+velar]) \gg \textsc{Ident}([+strident]) \gg \\ & \gg \textsc{Ident}([+cons][-voice]) \gg \textsc{Ident}([+cons]) \gg \\ & \gg \textsc{Faith} \gg * \begin{bmatrix} +cons \\ -voice \end{bmatrix} \begin{bmatrix} +cons \\ +voice \end{bmatrix} \gg \\ & \gg \textsc{Dep}([-cons]) \gg \textsc{Max}([+cons]) \end{cases}$$

Description length: $|G_{initial}| + |D{:}G_{initial}| = 1,059 + 4,000 = 5,059$

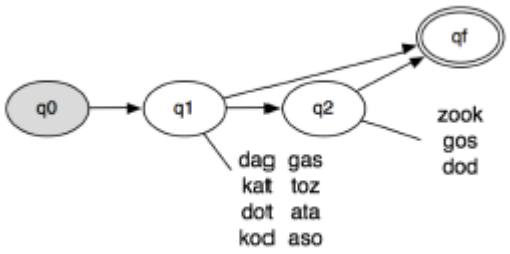
Figure 3.3: Voicing assimilation: initial grammar

The data corpus was multiplied by 25 in order to save on the running time of the simulation. A simulation with corpora containing a large number of surface forms or with realistic language corpora would increase the running time dramatically, therefore we introduced the data multiplication factor to be able to obtain the results within a shorter time frame. $D|G$ is affected by the amount of data we present to the learner, and the more surface forms we have in the corpora, the higher the number of bits and the longer the running time of the algorithm will be. The initial temperature was set to 200, and the cooling parameter was 0.99995. The goal of the simulation was to learn the morphemes and produce a new lex-

icon HMM with stems and suffixes separated, as well as to infer the devoicing process at the morpheme boundary and the constraint hierarchy supporting this process.

Final grammar:

$$G_{final} = \begin{cases} \textsc{Lex}: \\ \\ \\ \\ \\ \\ \\ \textsc{Con}: \quad \textsc{Max}([+cons]) \gg \textsc{Dep}([-cons]) \gg \\ \qquad \gg * \begin{bmatrix} +cons \\ -voice \end{bmatrix} \begin{bmatrix} +cons \\ +voice \end{bmatrix} \gg \textsc{Faith} \gg \\ \qquad \gg \textsc{Ident}([+cons]) \gg \textsc{Ident}([+cons][-voice]) \gg \\ \qquad \gg \textsc{Ident}([+strident]) \gg \textsc{Ident}([+velar]) \gg \\ \qquad \gg \textsc{Ident}([-strident]) \gg \textsc{Ident}([-velar]) \end{cases}$$



Description length: $|G_{final}| + |D|G_{final}| = 432 + 4,400 = 4,832$

Figure 3.4: Voicing assimilation: final grammar

During the search, the surface forms were successfully decomposed into stem and suffix morphemes. As shown in Figure 3.4, the final lexicon consists of stems and suffixes, which are represented as HMM emissions in states q_1 and q_2 respectively, as well as the transitions between the stem and suffix states: $q_1 \rightarrow q_2$ (stem-to-suffix transition) and $q_1 \rightarrow q_f$ (stem-to-final-state optional transition,

which doesn't involve the suffix). The overall description length ($D|G$) decreased from 5,059 to 4,832 ($\approx 5\%$), which may not seem as a drastic decrease, however, by examining the description lengths of the grammar components we can observe that the initial G description length decreased by 40% (from 1,059 to 432), while the D description length increased by 10% (from 4,000 to 4,400). The successful result was achieved after running multiple simulations (over 80) and stopping some of them when the outcomes did not look promising.

The learner started with a reversed set of constraints, under which no generalization regarding the voicing assimilation in data could be derived, the parsed URs would contain both $[+voice]$ and $[-voice]$ morpheme-final consonants (e.g., the UR to SR parses would contain both $/dotzook/ \rightarrow [dotzook]$ and $/dotsook/ \rightarrow [dotsook]$), and hence describing the data given the initial constraint set required more bits. The learner succeeded in deriving the voicing assimilation process by selecting the final constraint set which is able to parse the optimal URs given the data – there are no $[-voice]$ morpheme-final consonants in the stems, the suffix pairs like $[sook]$ and $[zook]$ are collapsed into $[zook]$, and the UR to SR parses are as follows: $/dotgos/ \rightarrow [dotkos]$; $/dotzook/ \rightarrow [dotsook]$; $/atadod/ \rightarrow [atadod] \ldots$, etc. This generalization allows the learner to make fewer choices when specifying a surface form by storing less forms in the lexicon and decreases the description length of the grammar by almost half.

As to the grammar component D, its description length increased from 4,000 to 4,400, and the cause of this increase is the stem-suffix decomposition and a need to represent the suffixes in a separate HMM state for the purposes of cor-

rect segmentation and morpheme order. Additionally, when the data words are segmented into morphemes, there is a need to add a transition between the stem morpheme to the suffix morpheme, as well as the optional transition from stem to the final state of the HMM (since this simulation is modeled after plural English devoicing, the devoicing happens only when a plural suffix is adjoined to a singular stem, and a transition from stem to the final state without changing to the plural form, i.e. having a null suffix, is also legitimate). Given the additional state and transitions, the description of data increases, at the same time allowing for the learner to infer the correct morpheme segmentation and order.

3.2 Inter-phonemic and inter-morphemic epenthesis

This simulation demonstrates the learner's ability to jointly learn morphology, phonology, and the application of morphophonological constraints between the morphemes and phonemes based on unanalyzed surface forms. The learner is presented with a list of surface forms generated on the basis of the alphabet $\sum = \{a, b\}$ and a feature table with one feature $\pm cons$ ($a = [+cons], b = [-cons]$), which start either with a null prefix or with the prefix *aab-*. The task of the learner in this simulation is to learn the distinction between the prefix *aab-* and the stems, the morpheme ordering of the forms, and the epenthesis of "*a*" between the "*bb*" sequences both in the stems and at the prefix-stem boundaries.

In this simulation, the initial constraint set (Figure 3.7) contains only one

Faithfulness constraint FAITH, which enforces identity mappings between the UR morphemes and surface forms. As opposed to the task of inferring an optimal constraint set for surface form generation by permuting the order of the constraints initially presented to the learner in reverse, having only one Faithfulness constraint poses a possibly harder task for the learner by giving it more freedom of choice. Here, the learner's goal is to infer the correct constraint hierarchy applicable to epenthesis, while starting the search without any Markedness constraints presented to it in advance. The initial HMM (Figure 3.7) contains all of the surface forms in its single inner state q_1. Figures 3.5 and 3.6 show the feature table and the data morphemes respectively.

	$cons$
a	$-$
b	$+$

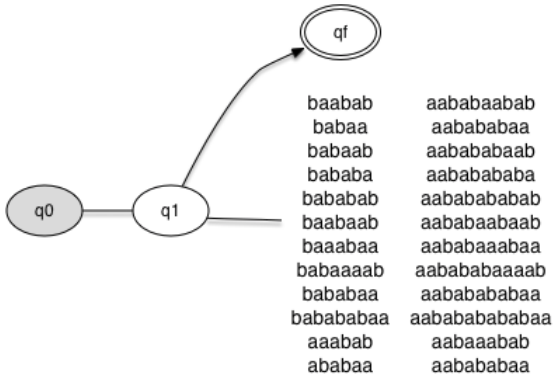
Figure 3.5: Complex morphophonology feature table

Prefix	Prefix + Stem
$\emptyset$	*ab, ba, bab, aba, abab, baa, baab* ...
aab	*aabaab, aabab, aababa, aababab, aababaa* ...

Figure 3.6: Complex morphophonology corpus stems and prefixes

Initial grammar:

$G_{initial} =$ { LEX:



CON: FAITH

Description length: $|G_{initial}| + |D{:}G_{initial}| = 497 + 3,000 = 3,497$

Figure 3.7: Complex morphology: initial grammar

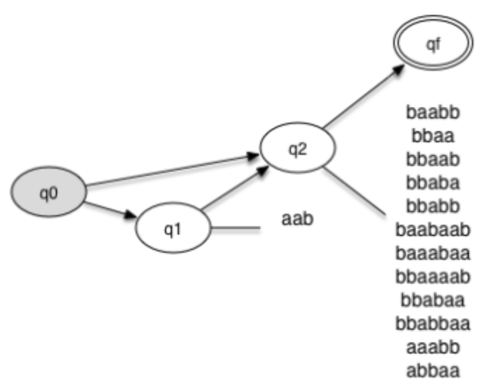
As in the previous simulation, the data was multiplied by 25 due to search performance considerations. The initial temperature was set to 100 and the cooling rate was set to 0.99995. In this simulation no segment changes were allowed. The learner successfully discovered the absence of *bb* sequences in the stems, as well as between the prefix and the stems, arriving at a more compact and restrictive description of the grammar. This demonstrates that the learner was able to induce morphophonological grammar in the absence of surface form alternations, as well as infer that the hypothesis encoding this pattern is more favorable as opposed to a hypothesis, which would treat the absence of *bb* as an accident. Also, the learner managed to infer a constraint hierarchy applicable to both inter-morphemic and inter-phonemic epenthesis, and, as seen in Figure 3.8, all relevant instances of "*a*"

are no longer in the lexicon. We ran 16 simulations in total, and the successful result was achieved only in one simulation.

The overall description length decreased from 3,497 to 3,316, as shown in Figure 3.8. The description length of G decreased $\approx 30\%$ from 497 to 316 – although the length of constraint set has increased from one faithfulness constraint to five Markedness constraints, it contributed to the decrease of the description length. The Markedness constraints derived by the learner allow it to make the lexicon more compact by preventing the "*bb*" sequences to appear in surface forms. The FAITH constraint prevents the epenthetic "*a*" from incurring more violations than the deletion of "*b*".

Final grammar:

$G_{final} =$ LEX:



CON: $* \begin{bmatrix} +cons \\ -voice \end{bmatrix} \begin{bmatrix} +cons \\ +cons \end{bmatrix} \gg \text{MAX}([-cons]) \gg \text{MAX}([+cons]) \gg \text{DEP}([-cons]) \gg \text{DEP}([+cons]) \gg \text{FAITH}$

Description length: $|G_{final}| + |D|G_{final}| = 316 + 3,000 = 3,316$

Figure 3.8: Complex morphology: final grammar

The description length of D remained the same $-$ 3,000, however, compared with the initial lexicon in Figure 3.7, the bits required to describe the data are distributed in a different manner: the initial lexicon HMM derived from the data is a single-state HMM, while the final lexicon is represented by the HMM with the prefix state q_1 and the stem state q_2, and the prefix-to-stem and initial-to-stem state transitions, the latter being the transition that occurs given the null prefix. The final lexicon successfully separates the prefix from the stems and preserves the correct morpheme ordering, and the constraint set ensures the compression of the lexicon by supporting the epenthesis process. The derived URs no longer contain epenthetic *a*'s at the morpheme boundaries and in the stems, and the parsed forms are as follows: $/bbaa/ \rightarrow [babaa]$; $/aabbbaa/ \rightarrow [aababababaa]$.

3.3 Vowel harmony

This simulation is a preliminary investigation in vowel harmony acquisition based on a 4-letter alphabet. Our goal was to show the general idea behind the process, while we hope to develop the full demonstration of vowel harmony acquisition in the future. The learner's task was to acquire the vowel harmony constraints for surface forms with the [$\pm$ back] trigger vowel in the stem and the target vowel in the suffix, in addition to learning the morpheme ordering and segmentation.

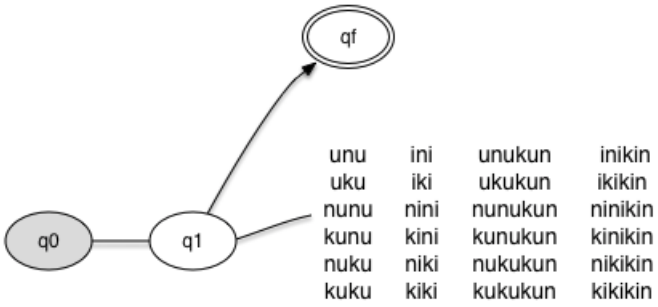
The learner was presented with 24 surface forms, consisting of combinations of 12 stem and 2 suffix morphemes (including the $\emptyset$ suffix). As in the voicing assimilation trial, the initial constraint set was a reversed version of a vowel harmony

enforcing constraint hierarchy, and GEN was allowed to change alphabet segments besides the regular insertion and deletion mutations. Enforcing the vowel harmony was done via phonotactic constraints, which penalize $V_{[+back]}CV_{[-back]}$ and $V_{[-back]}CV_{[+back]}$ sequences. The initial HMM (Figure 3.9) had a single inner state q_1, the emissions of which were the words extracted from the corpus. Figures 3.10 and 3.11 show the feature table and the surface form morphemes.

Initial grammar:

$G_{initial} =$

LEX:



CON: $\text{FAITH}([-velar]) \gg \text{IDENT}([-cons]) \gg$
$\gg \text{IDENT}([+cons]) \gg \text{DEP}([+cons]) \gg$
$\gg \text{DEP}([-cons]) \gg \text{MAX}([+cons]) \gg$
$\gg \text{MAX}([-cons]) \gg$
$\gg * \begin{bmatrix} -cons \\ +back \end{bmatrix} [+cons] \gg \begin{bmatrix} -cons \\ -back \end{bmatrix} \gg$
$\gg * \begin{bmatrix} -cons \\ -back \end{bmatrix} [+cons] \gg \begin{bmatrix} -cons \\ +back \end{bmatrix} \gg$
$\gg \text{DEP}([-cons]) \gg \text{MAX}([+cons])$

Figure 3.9: Vowel harmony: initial grammar

	$cons$	$back$
i	$-$	$-$
u	$-$	$+$
n	$+$	$-$
k	$+$	$-$

Figure 3.10: Vowel harmony feature table

Stem	Suffix
unu, uku, nunu, kunu, nuku, kuku,	$\emptyset$
ini, iki, nini, kini, niki, kiki	kun

Figure 3.11: Vowel harmony corpus stems and suffixes

In this preliminary investigation we ran 12 simulations with the data multiplied by 25, the initial temperature set to 100 and the cooling rate set to 0.99995. Although the learner was unable to acquire morphological segmentation at this stage, the phonological learning results were interesting – in most simulations, the learner's preferred constraint set was as follows:

$*[[+back, -cons][+cons][-back, -cons]] \gg$ DEP$[+cons] \gg$

$\gg$ MAX$[+cons] \gg$ MAX$[-cons] \gg$ DEP$[-cons] \gg$

$\gg *[[-back, -cons][+cons][+back, -cons]] \gg$

$\gg$ IDENT$[-cons] \gg$ IDENT$[+cons] \gg$ FAITH

Under this constraint hierarchy, the learner inferred the vowel harmony on the phonological level. The UR $\rightarrow$ surface form parses were observing the intended vowel feature spreading, however the VCV sequences in the URs derived by the learner would follow the $V[-back]CV[+back]$ or $V[+back]CV[-back]$ pattern, thus allowing for optionality in spreading the features, as shown in Figure 3.12:

No.	UR	SR
1	inu →	ini, unu
3	kiku →	kiki, kuku
4	kinu →	kini, kunu
5	niku →	niki, nuku
6	ninu →	nini, nunu
7	unikin →	inikin
8	ukukin →	ukukun
9	kikukin →	kikikin
10	kinikun →	kinikin
11	kukikun →	kukukun
12	kunukin →	kunukun
13	nikikun →	nikikin
14	ninukun →	nunukun

Figure 3.12: Vowel harmony UR → SR parses

The parses in $1-6$ demonstrate that under the constraint set preferred by the learner, the UR → SR mappings produce two outputs, i.e., a UR like "inu" maps to two surface forms, "ini" and "unu". In $7-14$, the [±back] feature spreads either to the first, middle, or last vowel of the form. Although the vowel harmony is being observed under the constraint set, and the vowels in surface forms harmonize, the URs inferred by the learner are not the expected result. Nonetheless, these results seem promising, and we hope to approach vowel harmony learning in future work.

Chapter 4

Previous learning models

In this chapter we will examine previous learning approaches proposed in the literature within the framework of OT. We will present three prominent approaches and review them regarding the evaluation metric in terms of economy and restrictiveness. In Section 4.1 we describe the family of paradigm-based learners, which focused on particular learning problems, as opposed to all-encompassing solutions proposed in probabilistic approaches and MDL. Sections 4.2 and 4.3 describe the models with a closer approach to MDL − the Maximum Likelihood Learning of Lexicons and Grammars (Jarosz, 2006), and the Lexical Entropy Learner (Riggle, 2006a). These two models target the economy and restrictiveness criteria, however, the imbalance between these criteria leads to challenges mentioned in Chapter 1 and further discussed below.

4.1 Paradigm-based lexicon learners

The guiding principles in OT have been the *Richness of the Base* (ROTB; Prince and Smolensky, 1993; Smolensky, 1996) and *Lexicon Optimization* (Prince and Smolensky, 1993). As Tesar and Smolensky state, ROTB has significant implications for the restrictiveness of the grammar, in particular, the relationship between the Markedness and Faithfulness constraints – for underlying structures with marked violations to appear in surface forms, Faithfulness must dominate the violated Markedness constraints. By observing the alternations between the underlying and surface forms a learner can select the correct hierarchy, since alternations occur when the UR is altered to satisfy high-ranked Markedness constraints at the expense of Faithfulness, and thus, alternations can be used as evidence that Faithfulness constraints are dominated. In the absence of alternations, it is proposed to place an inductive bias on the *initial constraint hierarchy* presented to the learner, where Markedness constraints dominate Faithfulness constraints (Alderete and Tesar, 2002; Tesar and Smolensky, 2000).

Tesar and Smolensky provided a useful foundation for exploring learnability in OT with their Constraint Demotion family of algorithms, namely the *Recursive Constraint Demotion (RCD)* (Tesar and Smolensky, 2000) and *Biased Constraint Demotion (BCD)* (Tesar and Prince, 2003).

The basic idea behind Constraint Demotion is that the learner, not being informed about the correct ranking by positive data in isolation, is presented with $\langle input, output \rangle$ pairs to select the "winner" (the most harmonic candidate) among

the competing candidates. The learner's task is to determine the constraint ranking based on the surface forms of the language. To do this, it has to compare each output candidate with the rest, and determine the dominant constraints – if the optimal output violates certain constraints, they must be dominated by another constraint in order to exclude the sub-optimal candidate. The constraints violated by the chosen candidate are demoted in the constraint hierarchy to be on a stratum that is immediately below the highest-ranking constraint which penalizes the sub-optimal candidate, so that the chosen candidate is the one "least offensive" to the constraints. By comparing each of the "winner-loser" pairs, the correct ranking makes one candidate more harmonic than its competitor, and the *dominance hierarchy* with the harmonic ordering of constraints is formed. An example of competing candidates is shown in Figure 4.1. The first candidate, '*a*', violates constraints C_2 and C_3, and candidate '*b*' violates C_4. Given the unranked initial hierarchy $\{C_1, C_2, C_3, C_4\}$, the constraint C_2 will be demoted below C_4, resulting in $\{C_1, C_3, C_4\} \gg \{C_2\}$, and C_3 will be demoted below C_4, resulting in $\{C_1, C_4\} \gg \{C_2, C_3\}$.

	C_1	C_2	C_3	C_4
☞a		*	*	
b				*

Figure 4.1: Constraint demotion tableau

In the case when the candidates have common violation marks, *Mark Cancellation* is applied − the constraint violation marks of "winner − loser" data pair are being compared and the common marks are canceled, which leaves only the

candidates with the uncanceled marks for comparison, and it is assumed that the marks of the sub-optimal candidate must be collectively worse than the marks of the optimal candidate. If two or more candidates are equally harmonic, and both are more harmonic than all the other candidates, both of them are optimal with the interpretation of free alternation.

CD depends critically on the assumption that a target language is given by a totally ranked hierarchy. When presented data from a non-totally ranked stratified hierarchy, it is possible for CD to go into an infinite loop. For example, given constraints *C1* and *C2* and the candidate parses *p1* and *p2*, when the constraints are located in the same constraint hierarchy layer, e.g. $\{C1, C2\} \gg C3 \gg \{C4, C5\}....Cn -$ if *p1* violates *C1* and *p2* violates *C2*, then, when the learner observes *p1*, it will infer that *p2* is the loser, and will demote *C1* below *C2*. Upon observing *p2*, the learner will infer that *p1* is suboptimal, it will demote *C2* below *C1* and run endlessly within the loop of demotions (Tesar and Smolensky, 2000).

RCD and BCD were not to be taken as learning algorithms on their own, but instead were intended as a component of an online learning procedure. These algorithms dealt primarily with the phonotactic learning, based on the observation that it occurs prior to the learning of morphology. Within the framework of the BCD algorithm, which was a further extension of RCD, the learner would receive information about morphological composition of each surface form, and use the phonotactic grammar induced in the first stage as an aid for UR search and acquisition. The morphological mapping would restrict the UR search in such a way that distinct surface forms of morphemes must be derived from identical URs.

During the phonological learning, the $\langle input, output \rangle$ pairs, where the input is identical to the output, are provided to the learner. Then, the learner would proceed with the BCD algorithm in search of a grammar, which is the most restrictive and consistent with the output forms. The restrictiveness of the grammar is estimated in accordance with the *r-measure* $-$ a criterion based on the $M \gg F$ constraint ranking (the r-measure of a language is calculated by counting, for each Faithfulness constraint *F*, the number of Markedness constraints *M* that dominate *F*). The higher the r-measure, the more restrictive the grammar. The result of the phonological learning stage would be a phonological grammar, which presents the optimal relative ranking of its Markedness constraints (when Markedness will always be ranked higher than Faithfulness), without inferring the relative ranking between the Faithfulness constraints, since they are not violated in the training set.

During the morphological learning, the learner is presented with alternation-based data together with the morphological structure of the language. The task of the learner is to construct possible URs by finding the alternating features and composing them into various combinations. Presented with an alternating surface pair, Lexicon Optimization principle will choose the UR that has fewer violations. If the morphemes do not alternate (or there is no pair), the UR is considered identical to surface form. Tesar and Smolensky note, that Lexicon Optimization needs to be applied not to individual forms, but to entire paradigms. In the end of the process, the hypothesis which is the most consistent with the grammar, is adopted.

The paradigm-based learning approach was primarily concerned with developing a provably correct algorithm for restrictive grammars, and apart from the

alternation-based phonological and morphophonological learning, the proposed learners offered no solutions regarding non-alternating URs with non-identical surface mappings. Although the alternations provide important information during language acquisition, they are a special case among the non-alternating structures of a language.

Alderete and Tesar (2002), McCarthy (2005), and Krämer (2012) attempted to address the challenge of the paradigm-based learners in acquiring non-identical URs in the case of non-alternating forms, and suggested modifying them in order to learn non-identical mappings from non-alternating URs. However, the proposed solutions were partial at best.

McCarthy (2005) demonstrated evidence from Choctaw, Japanese, Rotuman and Sanskrit, where some non-alternating URs are distinct from their surface forms, and suggested to extend the non-identical mappings in alternating forms to non-alternating forms by introducing the Free-Ride principle. According to Free-Ride approach, a learner, presented with alternations in which *some* surface forms are derived from specific URs, will generalize and derive *all* surface forms from these URs, including the non-alternating surface forms. Therefore, allowing non-alternating surface forms to take a "free ride" on the $/A/ \rightarrow [B]$ unfaithful map must resolve the issue and achieve a consistent and a more restrictive grammar with a smaller lexicon compared to the grammar obtained by an identity map. However, the Free-Ride principle does not provide a solution and even proves to be redundant for cases when alternating surface forms are derived from more than one UR, and does not work well with contextually restricted free rides (Mc-

Carthy, 2005). Nevins and Vaux (2007) showed that based on the examples of Spanish rhotics, where the contrast between the flap (‘ɾ’) and the trill (‘r’) is neutralized since only the trill is possible in word-initial position, the UR phonemes will result in an unfaithful identity map:

(17) Representation of [rosa] (surface trill)

a. /rosa/ (underlying trill)

b. /ɾosa/ (underlying flap) undergoes initial trilling due to surface word-initial constraint: *ɾ

Naturally, the analysis in item (b) is more complex. In the absence of alternation, there would be no reason to resort to (b) at all − the ROTB would consider the UR phonemes as non-alternating cases and Lexicon Optimization would only consider the UR with the faithful mapping. There is also no Free Ride to be applied here, since there are no alternations which would turn the underlying flap into a trill in the beginning of a word. Nevins and Vaux then present a process of turning an initial rhotic into a non-initial segment of the word, based on a language game that inverts the order of syllables (“casa” becomes “sa.ca”, “gato” becomes “to.ga”, etc.). If applied to the above example 17, and assuming that “rosa” is stored with a UR that contains a flap, the initial rhotic will become a non-word-initial flap:

(18) a. [ɾosa] → [saɾo]

b. /ɾosa/ → [saɾo]

Otherwise, if the UR of "rosa" is stored with a trill, there is no rule forbidding non-initial trills. However, an assumption that learners invent a rule based on this language game would be very far-fetched. Therefore, relying on Free-Ride (as well as on ROTB or Lexicon Optimization) to determine which UR is stored and picked would not help.

Nevins and Vaux (2007) also showed that there are many cases where LO is not respected in the absence of alternations, and that UR construction complexity goes beyond just relying on tableaux construction. By analyzing nonce word production in Turkish and Dutch speakers, they state that morphological knowledge, lexical statistics, segmental frequencies and orthographic representations all play a role in UR constructions, and that these factors consistently overweigh the Lexicon Optimization procedure. In this respect, it is safe to assume that a model relying on constraint re-ranking and Lexicon Optimization would not be able to take into account the influence of the factors described by Nevins and Vaux and would overgeneralize. Nevins and Vaux conclude that abandoning Lexicon Optimization would require the reassessment of Tesar and Smolensky's constraint demotion algorithm.

Krämer (2012) discussed the issues of ROTB, LO, and Free-Ride, which arise during the learner's inference of non-identical mappings for non-alternating forms, and suggested that a development of some form of the Free-Ride algorithm combined with the *mirror-image evaluation* of LO tableaux in order to remove the redundant features from URs and to keep the lexicon free from non-contrastive segments and features might help with decomposition of morphologically com-

plex forms.

Alderete and Tesar (2002) stated that although it has been a standard assumption that alternations are the only type of data that could possibly motivate lexical representations which are distinct from surface forms, some lexical aspects may be learned without seeing alternations in morphology/phonology. If important aspects of the phonological structure are not observable in the surface forms, the learner overgeneralizes, and the cases of phonological structures, which are not directly accessible in the output form can be demonstrated in the absence of alternations. With the evidence of stress and epenthesis interaction from Yimas, Mohawk and Selayarese, they demonstrated that BCD approach to lexical acquisition indeed leads the learner to favor superset grammars, and they proposed modifying any constraint-reranking learner so that it can acquire non-identical mappings from surface forms to the URs without alternations. Their suggestions included revising Prince and Tesar's *r-measure* in a way that will make it possible to represent a bias for the more restrictive grammar; or having a similarity metric for making generalizations over the lexicon (Frisch, 1997), which can serve the learner as a background for setting up the correct LRs for stress-epenthesis interaction.

Although various extensions for paradigm-based learners have been proposed, a constraint re-ranking learner, which utilizes these extensions in order to properly generalize beyond alternations still remains a task for future research.

4.2 Probabilistic models – MLG

Probabilistic versions of OT learning provided a more feasible approach to the subset problem – the evaluation metrics were better defined, and the models supported learning non-identical mappings of non-alternating forms. The Maximum likelihood learning of Lexicons and Grammars model (MLG), proposed by Jarosz (2006), is a probabilistic learner, which, instead of using the constraint ranking biases to achieve restrictiveness, treats a hypothesis as a distribution over constraint rankings and a distribution over each morpheme's UR. The search starts with an uncommitted lexicon, which contains unstructured surface forms without any prosodic or morphological structures associated with a sequence of morpheme indices, word frequencies, and uniform distribution over the space of possible URs for each morpheme. The grammar is represented as a probability distribution over total constraint rankings, and it defines a distribution over structured phonological forms for any UR. The goal of the search is to maximize the likelihood of the data with the help of Expectation Maximization algorithm (Dempster et al., 1977), on the basis of the set of constraints and UR candidates for each morpheme provided to the learner.

Learning in MLG relies on ROTB principle and likelihood maximization, where likelihood maximization defines the correct grammar and lexicon combination as the one that maximizes the likelihood of the surface forms. Let us demonstrate the MLG process for a variant of ab-nese based on the discussion from Rasin and Katzir (2016). The likelihood measure for a lexicon with the

surface forms from a variant of ab-nese, as in Figure 3.6, will be calculated as follows:

(19) If the morpheme is M_1, the likelihood of the surface form *ab* given that the featural variant for *a* is *e* and for *b* it is *p*:

$$P(surface = ab|M_1) = \sum_{u \in \{ab;ap;eb;ep\}} P(surface = ab|u)P(u)$$

With the set of morphemes $\{ab, ap, eb, ep\}$ where *p* is a featural variant of *b* and *e* is a featural variant of *a*, the initial probability distribution will be uniform:

(20)

$$M_1 = (ab); URs : ab(.25), ap(.25), eb(.25), ep(.25)$$

In order to calculate the likelihood of *ab* expressing the morpheme M_1, the four URs in 20 are enumerated and the conditional probability of the surface form *ab* is computed for each of the URs. The final result will be the weighted sum of conditional probabilities for *ab*, which is calculated by looking at different constraint rankings and their probabilities. Let us look at the permutations of the constraint set $\{*ab, *p, \text{IDENT}\}$, calculate the ranking probability, and observe the optimal morpheme outputs under each constraint hierarchy given input *ab*:

(21)

Hypothesis H		Probability under input *ab*	
Ranking r_i		$P(r_i)$	Optimal O_k
r_1	*ab ≫ *p ≫ IDENT	0.2	eb
r_2	*ab ≫ IDENT ≫ *p	0.15	eb
r_3	IDENT ≫ *ab ≫ *p	0.05	ab
r_4	*p ≫ *ab ≫ IDENT	0.1	eb
r_5	*p ≫ IDENT ≫ *ab	0.0	ab
r_6	IDENT ≫ *p ≫ *ab	0.5	ab

The sum of conditional probabilities of constraint rankings where *ab* is the winner, is $P(r_3) + P(r_5) + P(r_6) = 0.55$. The computation for other URs is performed in the same manner, and then the candidate with the maximum likelihood wins.

Based on the example above, we can see that only the forms that occur under a hypothesis get probability distribution values, which prevents overgeneration and makes the grammar restrictive. This is similar to $D|G$ minimization during the MDL-based learning. However, we can also observe that in order to encode the data, these hypotheses heavily rely on constraints and not on the lexicon, which is similar to ROTB. Even though the search starts with an uncommitted lexicon, which may serve as a proxy for the economy criterion, given special cases in the data, the learner's search procedure may result in overfitting, or, memorizing the data, rather than favoring more compact hypotheses.

Following Prince and Tesar (2004), Tesar and Prince (2003), Hayes (2004), the MLG learning model is realized in two stages − phonotactic and morpho-

phonemic learning:

1. Phonotactic Learning
    (a) A fixed, universal rich base is assumed
    (b) No morphological awareness
    (c) Grammar learning but no lexicon learning
2. Morphophonemic Learning
    (a) Words are analyzed into component morphemes
    (b) Learning of morpheme specific underlying forms occurs
    (c) Further learning of the grammar to account for alternations

The phonotactic stage consists of gradual learning of a grammar that maximizes the likelihood of the surface forms, given a (fixed) rich base. The rich base is an unbiased distribution over phonological forms, represented by the free combination of the phonological elements. Phonotactic learning results in a restrictive grammar that matches the frequencies of the surface forms. During morphophonemic learning the grammar and lexicon combination that maximizes the likelihood of the overt forms is gradually learned.

Following Rasin and Katzir's discussion of MLG, if we take the data from our complex morphology simulation given the constraints $\{*ab, *p, \text{IDENT}\}$, the hypothesis in (23) will receive the highest possible score during the phonotactic acquisition stage of the learner:

(22) a. M_1 =(ab) URs: *ab* (1); *ap* (0); *eb* (0); *ep* (0)

b. M_2 =(bab) URs: *bab* (1); *bap* (0); *beb* (0); *bep* (0); *pap* (0); *pep* (0); *pab* (0); *peb* (0)

c. M_3 =(abaa) URs: *abaa* (1), *apaa* (0), *epaa* (0), *epea* (0) ...

d. M_4 =(baaba) URs: *baaba* (1) ...

e. M_5 =(aabab) URs: *aabab* (1) ...

f. M_6 =(aababab) URs: *aababab* (1) ...

g. M_7 =(aababaa) URs: *aababaa* (1) ...

h. M_8 =(aababaaba) URs: *aababaaba* (1) ...

(23) IDENT $\gg *ab \gg *p$

The hypothesis above will be considered optimal under the maximum likelihood criterion. However, already at the phonotactic stage of learning, it is obvious that the hypothesis memorized the data − if we can list surface forms with a probability 1, the MLG will result in memorization of the data, where no generalization has been made, and fully memorized hypotheses will get a likelihood of 1 on the basis of faithful mappings dominating over markedness. The learner will not see any advantage in detecting the epenthesis of *a* between sequences of *bb* and describing this case via constraints. The absence of *p* is not taken into account as well. The faithfulness constraints will always be optimal in the presence of hypothesis where surface forms can be assigned a probability of 1. This serves as evidence that this learner, although starting with an uncommitted lexicon, ends up

relying on restrictiveness alone, which prevents it from favoring smaller lexicons and generalizing over subsets of possible forms rather than making narrower generalizations, by simply memorizing the data and not making any effort to describe this information via constraints.

Given the case above, what will happen during the morphophonemic learning stage? The learning will begin with the hypothesis represented by the memorized lexicon in (22) and the inferred constraint set in (23). The first task during the morphophonemic learning stage is to analyze the words into component morphemes. Since the **bb* constraint is not present in the winning hypothesis for the phonological part, the prefix may be identified as "aab" or as "aaba". Regardless of that, the constraint enforcing inter-morphemic "b.b" epenthesis is not present, and will not be learned. At the second stage of learning morpheme specific URs, in the absence of an epenthesis constraint, the stems will remain as they were memorized in the phonotactic step, the URs of either one or the other prefix will remain unchanged, no inter-phonemic epenthesis will be induced, and the model will remain overfitted. Needless to say, the third stage would not introduce anything new into the hypothesis.

By approaching restrictiveness directly, Jarosz's probabilistic formulation of ROTB suggests a solution to learning non-alternating URs. However, during the MLG learning process, the uncommitted lexicon presented to the learner only affects the beginning of the search, while the rest of the search relies on constraints. The learner sees no benefit in making the lexicon more compact, thus affecting the outcome for the optimal grammar hypothesis. Although MLG will

favor grammars which describe the data well, it will disregard their compactness. Based on the examples above we may conclude that in order to escape overgeneralization and overfitting, the economy criterion must be represented directly and continuously throughout the learning process.

4.3 Lexical entropy

Using an uncommitted lexicon only at the initial stage showed to be little help to the learner. However, the entropic property of a lexicon can contribute to grammar compactness and good hypotheses. Riggle (2006b) suggested using lexicon entropy as a learning criterion, stating that selecting the most entropic grammars is the direct implementation of the ROTB principle, where the set of possible inputs to grammar is universal. The phonology learner proposed by Riggle evaluates hypotheses using the *lexical entropy* measure (a compactness measure), which is based on making a decision whether to encode a phonological pattern as resulting from constraint interaction or as a special case in the lexicon. Whenever faced with this decision, the learner will choose the grammar that characterizes this pattern as a consequence of the constraints, rather than an accident. This strategy is based on the properties of the input sets that each candidate grammar (ranking hypothesis) associates with a given phonological pattern, and not on the formal properties of the constraint rankings themselves.

The evaluation of a grammar *G*, according to the conditional entropy of *G*'s lexicon is defined in terms of conditional entropy for bigrams:

(24)

$$H(G) = -\sum_{x\in\Sigma}\sum_{y\in\Sigma} P(x,y)logP(y|x)$$

The learner will prefer a hypothesis where ***H(G)*** is higher. Starting with the empty hypothesis space and the empty set of observed forms, the learner observes the first presented form and obtains a set of $\langle output, ERC\rangle$ pairs. ***Elementary Ranking Conditions*** (Prince, 2002) define a disjunction of partial constraint rankings under which a contender is more harmonic than the other contenders for the same input. Then, each set is evaluated based on the internally consistent set of constraint ranking statements to find the grammar that maps the most entropic, least restricted, set of inputs to the observed forms.

Comparing the results of deriving a correct hypothesis for the *ab-nese* corpus used in our complex morphophonology simulation, and following the discussion in Rasin and Katzir (2016), the lexical entropy learner would dismiss the IDENT hypothesis, and favor the hypothesis with the epenthesis-enforcing constraint ranking:

(25) Hypothesis A (identity)

Lexicon:

1)	/ab/	3)	/abaa/	5)	/aabab/	7)	/aababaa/
2)	/bab/	4)	/baaba/	6)	/aababab/	8)	/aababaaba/

CON: any

Entropy: 0.61

(26) Hypothesis B (correct)

Lexicon:

1)	/ab/	3)	/abaa/	5)	/aabb/	7)	/aabbaa/
2)	/bb/	4)	/baaba/	6)	/aabbb/	8)	/aabbaaba/

CON: $*bb$, MAX $\gg$ DEP

Entropy: 1

In (25) the lexicon URs are identical to the provided surface data, and all URs will remain unchanged under any ranking. The constraint **bb* will be considered an accident of the lexicon, and the predictions of this hypothesis will be based on the probabilities of the adjacent segments ($P(b|a) = 1.0$). In (26), the predictable information about **bb* sequences is removed, which makes predictions regarding adjacent segments harder. Thus, the entropy measure will be higher and the constraint ranking will be more restrictive, e.g. **bb* $\gg$ DEP.

Riggle suggests that entropy is the only factor in the learning criterion, which emphasizes the importance of compactness, however, this model lacks the requirement for restrictiveness. The absence of any pressure for restrictiveness leads to the subset problem. And, indeed, if the lexicon entropy learner is not provided with any constraint ranking in advance, it will result in a superset grammar. A hypothesis with no constraints to rank will be maximally entropic and will overgenerate. As discussed by Rasin and Katzir, given a lexicon with a uniform bigram distribution, such as /aabba/ (where $\forall(x, y)P(x|y) = 0.5$), and no constraints, the UR /aabba/ can be mapped to any form without any violation marks. This implies

that in order to achieve the best hypothesis, only the correct constraint ranking must be provided to the learner − otherwise, it will overgenerate.

Although Riggle's model utilizes the compactness criterion in a more effective manner than MLG (in the presence of correct constraint ranking only), it is evident that economy alone will not suffice in escaping the subset problem, it must be combined with a proper restrictiveness measure, and represented directly. In Riggle's learner, the compactness is represented via entropy, which tends to introduce disorderly material into the grammar besides removing the orderly material, as discussed in Rasin and Katzir (2016).

To conclude, the issues of the models described above serve as supportive evidence for the claim that economy and restrictiveness must be maximized together and represented directly in order to arrive to the optimal hypothesis. Following this principle, Rasin and Katzir's MDL model succeeds in generating both compact and restrictive grammars, as well as working with alternating and non-alternating corpora. Additionally, it allows to acquire phonology and morphology simultaneously, without having to divide the acquisition into stages, proving to be the simplest model thus far.

Chapter 5

Discussion

In this work, we presented the MDL-based computational model for unsupervised joint learning of morphophonological constraints and lexicons within the framework of OT. This model was developed to extend Rasin and Katzir's phonological learner in order to explore its abilities for acquiring both phonological and morphophonological constraints, as well as morpheme ordering in lexicons given unanalyzed artificial corpora with surface forms. Based on three simulations – a simulation modeled after plural English, a more complex morphophonology simulation involving inter-phonemic and inter-morphemic epenthesis constraints, and a preliminary investigation of vowel harmony, this MDL-based learner was able to:

1. Learn from unsupervised data;
2. Successfully induce lexicons and OT constraints applicable on inter-phonemic and inter-morphemic levels;

3. Arrive at phonological and morphological generalizations simultaneously during one learning process;

4. Converge on optimally compact and restrictive grammar hypotheses.

The model presented in this work suggests that MDL learning is, indeed, a viable and simple solution for morphophonological learning, which can lead to its further applications to a wider range of morphophonological issues, as well as to developing more elaborate, cognitively plausible learners based on the realistic language data.

The corpora presented to the learner within the framework of this research was small and artificial for the sake of simulation running times. Although we were able to achieve correct UR generation when testing a small, but realistic Tuvan corpus (see Appendix A), due to the running time limitations at the stage of the current research, this simulation would have taken more than several months to complete. In all simulations, the data was multiplied by a factor instead of presenting the learner with bigger corpora for the same reason, and developing an approach to work with bigger corpora containing realistic language forms could result in more interesting conclusions. The project was coded in Python, and attempting to implement it in C could speed up the simulation running times and possibly allow for using larger corpora.

In order to make the grammar components presented to the learner computationally viable, the constraint set and the lexicon were represented as automata. The constraint set FSTs were based on Riggle's finite-state OT model, and the

lexicon was represented by the HMM. For the purposes of output generation, the HMM was converted into a parsing NFA, where all possible paths are created based on the length of the longest word in data, and then the data parses are being evaluated. Since the transducer composition rules follow Riggle's definitions (in Riggle the constraint FSTs undergo intersection) and are different from the default state machine composition rules, it may be worthwhile and more efficient to extend these rules onto the HMM lexicon representation and attempt to compose the HMM directly with the constraint set FST, rather than rely on the longest word in data.

We have shown that MDL-based evaluation metric is able to induce lexicons, constraint rankings, and constraints with and without supporting data from alternations by jointly learning phonology and morphophonology. Considering the results of MDL-based learning, the current research contributes to modeling further learning processes within the MDL framework, trying out alternative metaheuristics, and developing further learners to be compared with the results of other language acquisition models.

Appendix A

Tuvan data

Corpus: maslo, maslolar, buga, bugalar, ygy, ygyler, teve, teveler, orun, orunnar, sivi, siviler

Feature table:

	son	*cons*	*back*	*round*	*high*
a	+	−	+	−	−
e	+	−	−	−	−
o	+	−	+	+	−
i	+	−	−	−	+
u	+	−	+	+	+
y	+	−	−	+	+
b	−	+	−	−	−
g	−	+	−	−	−
l	−	+	−	−	−
m	−	+	−	−	−
r	−	+	−	−	−
s	−	+	−	−	−
t	−	+	−	−	−
v	−	+	−	−	−
n	−	+	−	−	−

Constraint set:

$\textsc{Max}[-cons] \gg \textsc{Dep}[-cons] \gg$

$\gg *[[+back, -cons][+cons][-back, -cons]] \gg$

$\gg *[[-back, -cons][+cons][+back, -cons]] \gg$

$\gg \textsc{Ident}[-round] \gg \textsc{Ident}[-high] \gg$

$\gg \textsc{Faith} \gg$

$\gg *[[+cons][+cons]] \gg *[[-cons][-cons]]$

Parsing results:

maslo $\rightarrow$ maslo

maslolar $\rightarrow$ maslolar

buga $\rightarrow$ buga

bugalar $\rightarrow$ bugalar

ygy $\rightarrow$ ygy

ygylar $\rightarrow$ ygyler

teve $\rightarrow$ teve

tevelar $\rightarrow$ teveler

orun $\rightarrow$ orun

orunnar $\rightarrow$ orunnar

sivi $\rightarrow$ sivi

sivilar $\rightarrow$ siviler

Tuvan words were taken from Tuvan talking dictionary, Swarthmore college (http://tuvan.swarthmore.edu/).

Bibliography

Akers, Crystal Gayle. 2012. Commitment-based learning of hidden linguistic structures. Doctoral Dissertation, Rutgers University-Graduate School-New Brunswick.

Alderete, John, and Bruce Tesar. 2002. Learning covert phonological interaction: an analysis of the problem posed by the interaction of stress and epenthesis. Technical Report RuCCS-TR-72, Rutgers Center for Cognitive Science, Piscataway, NJ.

Angluin, Dana. 1980. Inductive inference of formal languages from positive data. *Information and Control* 45:117–135.

Apoussidou, Diana. 2007. *The learnability of metrical phonology*. LOT.

Baker, Carl L. 1979. Syntactic theory and the projection problem. *Linguistic Inquiry* 10:533–581.

Berger, Iddo. 2018. Unsupervised induction of rule-based morpho-phonology. MA Thesis in preparation, Tel Aviv University.

Berwick, Robert C. 1982. Locality principles and the acquisition of syntactic knowledge. Doctoral Dissertation, MIT, Cambridge, MA.

Booij, Geert. 2011. Morpheme structure constraints. In *The Blackwell Companion to Phonology. Volume 4. Phonological interfaces.*, ed. Elizabeth Hume Marc van Oostendorp, Colin J. Ewen and Keren Rice, chapter 86, 2049–2070. Oxford: Blackwell.

Braine, Martin D. S. 1971. On two types of models of the internalization of grammars. In *The ontogenesis of grammar*, ed. D. J. Slobin, 153–186. Academic Press.

Brent, Michael, and T. Cartwright. 1996. Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition* 61:93–125.

Calamaro, Shira, and Gaja Jarosz. 2015. Learning general phonological rules from distributional information: A computational model. *Cognitive Science* 39:647–666.

Chaitin, Gregory J. 1966. On the length of programs for computing finite binary sequences. *Journal of the ACM* 13:547–569.

Chater, Nick, and Paul Vitányi. 2007. 'Ideal learning' of natural language: Positive results about learning from positive evidence. *Journal of Mathematical Psychology* 51:135–163.

Chomsky, Noam. 1951. Morphophonemics of Modern Hebrew. Master's thesis, University of Pennsylvania.

Chomsky, Noam, and Morris Halle. 1968. *The sound pattern of English*. New York: Harper and Row Publishers.

Clark, Alexander. 2001. Unsupervised language acquisition: Theory and practice. Doctoral Dissertation, University of Sussex.

Dell, François. 1981. On the learnability of optional phonological rules. *Linguistic Inquiry* 12:31–37.

Dempster, Arthur Pentland, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* 39:1–38.

Dowman, Mike. 2007. Minimum description length as a solution to the problem of generalization in syntactic theory. Ms., University of Tokyo, Under review.

Frisch, Stefan. 1997. Similarity and frequency in phonology. Doctoral Dissertation, Northwestern University.

Goldsmith, John. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics* 27:153–198.

Goldsmith, John. 2010. Towards a new empiricism for linguistics. To appear as chapter 3 in Empiricist Approaches to Language Learning, co-authored with Alex Clark, Nick Chater, and Amy Perfors.

Grünwald, Peter. 1996. A minimum description length approach to grammar inference. In *Connectionist, statistical and symbolic approaches to learning for natural language processing*, ed. G. S. S. Wermter and E. Riloff, Springer Lecture Notes in Artificial Intelligence, 203–216. Springer.

Halle, Morris. 1962. Phonology in generative grammar. *Word* 18:54–72.

Hayes, Bruce. 2004. Phonological acquisition in Optimality Theory: The early stages. In *Constraints in phonological acquisition*, ed. René Kager, Joe Pater, and Wim Zonneveld, 158–203. Cambridge, UK: Cambridge University Press.

Hsu, Anne S., and Nick Chater. 2010. The logical problem of language acquisi-

tion: A probabilistic perspective. *Cognitive Science* 34:972–1016.

Hsu, Anne S., Nick Chater, and Paul M.B. Vitányi. 2011. The probabilistic analysis of language acquisition: Theoretical, computational, and experimental analysis. *Cognition* 120:380 – 390.

Inkelas, Sharon. 1995. The consequences of optimization for underspecification. In *Proceedings of NELS 25*, ed. Jill Beckman, 287–302. GLSA.

Jarosz, Gaja. 2006. Richness of the base and probabilistic unsupervised learning in Optimality Theory. In *Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology at HLT-NAACL 2006*, 50–59.

Katzir, Roni. 2014. A cognitively plausible model for grammar induction. *Journal of Language Modelling* 2:213–248.

Kirkpatrick, Scott, C. Daniel Gelatt, and Mario P. Vecchi. 1983. Optimization by simulated annealing. *Science* 220:671–680.

Kolmogorov, Andrei Nikolaevic. 1965. Three approaches to the quantitative definition of information. *Problems of Information Transmission (Problemy Peredachi Informatsii)* 1:1–7.

Krämer, Martin. 2012. *Underlying representations*. Cambridge University Press.

Li, Ming, and Paul Vitányi. 2008. *An introduction to Kolmogorov complexity and its applications*. Berlin: Springer Verlag, 3rd edition.

de Marcken, Carl. 1996. Unsupervised language acquisition. Doctoral Dissertation, MIT, Cambridge, MA.

McCarthy, John J. 2005. Taking a free ride in morphophonemic learning. *Catalan Journal of Linguistics* 4:19–56.

Merchant, Jason. 2008. An asymmetry in voice mismatches in VP-ellipsis and pseudogapping. *Linguistic Inquiry* 39:169–179.

Nevins, Andrew, and Bert Vaux. 2007. Underlying representations that do not minimize grammatical violations. In *Freedom of analysis?*, ed. Sylvia Blaho, Patrik Bye, and Martin Krämer, 35–61. Mouton de Gruyter.

Prince, Alan. 2002. Entailed ranking arguments. ROA500. Available at http://roa.rutgers.edu.

Prince, Alan, and Paul Smolensky. 1993. Optimality theory: Constraint interaction in generative grammar. Technical report, Rutgers University, Center for Cognitive Science.

Prince, Alan, and Bruce Tesar. 2004. Learning phonotactic distributions. In *Constraints in phonological acquisition*, ed. René Kager, Joe Pater, and Wim Zonneveld, 245–291. Cambridge University Press.

Pycha, Anne, Pawel Nowak, Eurie Shin, and Ryan Shosted. 2003. Phonological rule-learning and its implications for a theory of vowel harmony. In *Proceedings of the 22nd West Coast Conference on Formal Linguistics*, volume 22, 101–114. Somerville, MA: Cascadilla Press.

Rasin, Ezer, Iddo Berger, Nur Lan, and Roni Katzir. 2017. Learning rule-based morpho-phonology. Ms., MIT and Tel Aviv University. Available at http://ling.auf.net/lingbuzz/003665.

Rasin, Ezer, and Roni Katzir. 2016. On evaluation metrics in Optimality Theory. *Linguistic Inquiry* 47:235–282.

Riggle, Jason. 2004. Generation, recognition, and learning in finite state Optimal-

ity Theory. Doctoral Dissertation, UCLA, Los Angeles, CA.

Riggle, Jason. 2006a. Infixing reduplication in Pima and its theoretical consequences. *Natural Language and Linguistic Theory* 24:857–891.

Riggle, Jason. 2006b. Using entropy to learn OT grammars from surface forms alone. In *Proceedings of the 25th West Coast Conference on Formal Linguistics*, 346–353.

Rissanen, Jorma, and Eric Sven Ristad. 1994. Language acquisition in the MDL framework. In *Language computations: DIMACS Workshop on Human Language, March 20-22, 1992*, 149. Amer Mathematical Society.

Smolensky, Paul. 1996. On the comprehension/production dilemma in child language. *Linguistic Inquiry* 27:720–731.

Solomonoff, Ray J. 1964. A formal theory of inductive inference, parts I and II. *Information and Control* 7:1–22, 224–254.

Stolcke, Andreas. 1994. Bayesian learning of probabilistic language models. Doctoral Dissertation, University of California at Berkeley, Berkeley, California.

Tesar, Bruce. 2006. Faithful contrastive features in learning. *Cognitive Science* 30:863–903.

Tesar, Bruce. 2009. Learning phonological grammars for output-driven maps. In *Proceedings of NELS*, volume 39, 1013–0209.

Tesar, Bruce. 2014. *Output-driven phonology*. Cambridge University Press.

Tesar, Bruce, and Alan Prince. 2003. Using phonotactics to learn phonological alternations. In *Proceedings from the annual meeting of the Chicago Linguistic Society*, volume 39, 241–269. Chicago Linguistic Society.

Tesar, Bruce, and Paul Smolensky. 2000. *Learnability in Optimality Theory*. Cambridge, MA: MIT Press.