Learning Phonological Transductions from Sparse Data over Structured Representations

A Dissertation presented

by

Magdalena Markowska

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Linguistics

Stony Brook University

March 2026

Stony Brook University

The Graduate School

Magdalena Markowska

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation.

Jeffrey Heinz – Dissertation Advisor
Professor, Department of Linguistics

Owen Rambow – Chairperson of Defense
Professor, Department of Linguistics

Jordan Kodner
Associate Professor, Department of Linguistics

Jane Chandlee
Associate Professor, Department of Linguistics, Haverford College

This dissertation is accepted by the Graduate School

Eric Wertheimer

Dean of the Graduate School

Abstract of the Dissertation

Learning Phonological Transductions from Sparse Data over Structured Representations

by

Magdalena Markowska

Doctor of Philosophy

in

Linguistics

Stony Brook University

2026

This dissertation develops linguistically informed, data-efficient learning methods for morpho-phonological processes, showing that combining structured representations (features and tiers) with new inference mechanisms supports successful learning in small-data settings. A tension in computational phonology is that approaches with strong formal learnability guarantees, particularly within the grammatical inference tradition, often require highly specific characteristic samples that can be much larger than, and qualitatively unlike, the sparse and incomplete data available in realistic acquisition contexts. At the same time, other learning approaches that do perform well empirically do not clearly delimit what classes of processes they can learn or when they are guaranteed to succeed, making it difficult to connect experimental success to explanatory generalizations.

To bridge this gap, I first introduce Factor-by-Feature (FxF), a framework that integrates phonological features directly into grammatical inference by decomposing a mapping into feature-specific submappings over reduced alphabets. This factorization yields smaller inferred transducers and substantially reduces the data needed

for correct generalization in experiments with synthetic data. I then show that per-feature parameterization (including feature combinations, memory window size, and processing direction) can further reduce both the quantitative and qualitative characteristic sample requirements. Nevertheless, a learner with theoretical guarantees used for experiments in this setting (SOSFIA; (Jardine et al., 2014)) continues to struggle when tested on textbook-style datasets with missing forms. To address this, I introduce SEAL, a String Equation Adaptive Learner that treats subsequential transducer inference as enforcing globally consistent constraints on transition outputs. I show that within the FxF framework SEAL recovers correct generalization whereas SOSFIA could not. Finally, I develop an approach to tier discovery for long-distance processes (e.g., harmony) building on algebraic characterizations of phonological functions. Together, these results show that grammatical inference techniques, when paired with linguistically informed representations, can be used to learn both local and non-local morpho-phonological mappings from sparse data.

"All things are collection of atoms with spaces in between them, and atoms are in a given mathematical arrangement. This device breaks them down, stores them and then puts them together again in the same order. "

– The Twilight Zone, S4E3

Contents

List of Figures

List of Tables

Acknowledgements

Life is funny like that. One day, you find a Dunkin' Donuts napkin in the pocket of your skirt in Warsaw and decide to move to the United States; the next thing you know, you are writing the acknowledgments section of your dissertation.

I believe in structure, yet my life, academically and beyond, seems to have been built on seemingly instant decisions that simply felt right. And I think they were right. It all started at the University of Warsaw, when I took a phonology class and learned about underlying representations. I still cannot shake the awe I experienced then. Thank you, Professor Rubach, for being my first academic inspiration.

In my graduate program at CUNY, I was first introduced to programming, machine learning, and natural language processing. Had I not gone crying to my advisor about not being able to find linguistics in it all, I would not have been introduced to Jeffrey Heinz. Thank you for that, Kyle Gorman.

Thank you, Jeff, for always staying by my side—you have been a wonderful mentor who planted seeds of confidence in me. You are an inspiration at any and every level. My hope is to be that cheerleader for my future students the same way you have always been for us. I never would have thought that I would be working on computational pragmatics, but I guess everything is possible with Owen Rambow. Even getting married in a bar! I feel very lucky to have been invited to work with you after my first year at SBU. But I believe we had shared common ground before that.

Jane Chandlee, your work has been a cornerstone of this dissertation, and I have always admired the power that emanates from your humbleness. I feel proud to be within the local distance of k=10 from you among Jeff's academic children. Jordan Kodner: thank you for surviving my two QPs and the dissertation, and for always thinking about things I would not have thought about. They made my work that much better. Mark Aronoff: thank you for always being so cheerful and encouraging, and for always being ready to share an interesting story. Thomas Graf: thank you for always giving me the right advice, and for always being willing to give it. Anita Wasilewska:

thank you for teaching me mathematical thinking and for helping me survive the hardest years in the graduate program. We have truly found each other. Gary Mar: thank you for letting me experience another type of education. You have a very unique talent for creating a safe space to talk and think, a space where imposter syndrome fades away.

Han Li, Neda Taherkhani, Alina Shabaeva, Scotty Nelson: you have been my main support system during the past six years. I am not sure I would have done it without you—chances are that I wouldn't. John Murzaku, Adil Soubki, Peter Zeng: you will always be my conference dream team. I am also very grateful to many other friends who made this journey that much more memorable: Vinny Czarnecki, Daniel Greeson, Felix Fonseca, Salam Khalifa, Pardis Derakhshandeh, Smeet Chedda, Gaurav Verma, Andrija Petrović, Chelsey Dolinger, Yash Kumar, and Arya McCarthy. Thank you, Hossep Dolatian, for hair rings, and Alëna Aksënova, for being my substring buddy.

Annie Rudez, Duane Hosein, Edus Dander—thank you for being my NYC family and for always, always, always being by my side. I will always be by yours! Thank you, my "Polish mafia" family, for creating a sense of home for me: Ewelina Siembida, Karolina Siembida, Kasia Gałka, Ola Ryś. Daniel Cai: thank you for being my guardian angel. Szymon Miezianko, Maciej Skowera, Kinga Kuchcińska, Monika Radomska, Marcin Moskal, Sylwia Zańko, Małgosia Mazur, Tomasz Snopkiewicz, Dominika Drażyk—thank you for remaining my friends after I escaped the motherland and for always finding time to see me. Mamo, Tato, Ola i Kamil: dziękuję, że zawsze we mnie wierzyliście i że zawsze mogę na Was liczyć. Tortellini, Simone, and Frankie: miau miau hau. Kevinku Vu: Cheree, Cheree, my comic book fantasy, I love you!

Everyone I mentioned here, and all those who came my way and whom I may have forgotten to mention, have contributed to the person I am today. Life is funny like that sometimes: what feels like an instant decision may have had its own underlying structure all along.

For Tortellini and Simone...

Chapter 1

Introduction

Phonology is a formal theory of the sound patterns of a language, centrally concerned with how mental (underlying) representations are related to their phonetic realizations. This perspective raises two foundational questions. First, what is the nature of the phonological grammar—what properties define it, and what kinds of computations can it perform? Second, given a hypothesis about the form of the grammar, how can such an object be inferred from finite, positive evidence, and what inductive biases make that learning efficient?

This dissertation addresses the second question from a formal and computational perspective. In the generative tradition, a language can be treated as a formally characterizable and potentially infinite set of expressions made coherent and productive by a *finite* system (Chomsky, 1986). Adopting this view in phonology leads naturally to modeling grammars as computational devices that map underlying representations with encoded meanings to surface forms. From this standpoint, the learning problem is to infer such a device from limited exposure, which in turn requires specifying the representations used for inputs and outputs, delimiting a space of possible hypotheses, and providing a procedure for selecting an appropriate hypothesis given the data.

This way of organizing the learning problem has a long history in the literature. For example, Chomsky (1986, p. 30) argues that a child learning language must have:

i. "a technique for representing input signals;

ii. a way of representing structural information about these signals;

iii. some initial delimitation of a class of possible hypotheses about language structure;

iv. a method for determining what each such hypothesis implies with respect to each sentence; and

v. a method for selecting one of the (presumably infinitely many) hypotheses allowed by (iii) and compatible with the observed data."

Although Chomsky (1986) frames these requirements in terms of child language acquisition, the same architecture extends naturally to artificial learners of phonological grammars, especially when the artificial learner is exposed to learning conditions that approximate those faced by children: sparse and incomplete paradigms. Accordingly, this dissertation focuses on learning under such sparse exposure applying inductive biases that make such learning possible.

For (i–ii), we adopt the standard phonological view that segments are not the smallest explanatory units, and that phonological generalizations are typically stated over *features* and *natural classes* (Hayes, 2009; Zsiga, 2013; Bale and Reiss, 2018). One of the main contributions of this dissertation is a framework that enables grammatical inference learners for phonological mappings to operate over such *featural representations* rather than treating segments as atomic symbols. We show that featural representations shrink the learner's input alphabet, and therefore the size of the induced hypothesis representations, which, in turn, reduces the amount of data required for effective generalization.

For (iii), we begin with a deliberately restricted hypothesis space, motivated by the view that phonological processes are bounded by finite-state computation (Johnson, 1972; Kaplan and Kay, 1994). Since finite-state devices characterize the regular languages (and can be extended to model regular relations via transducers), this provides a natural upper bound on phonological computation. Nevertheless, the regular class is typologically too permissive. Recent work in *subregular* phonology argues that attested patterns occupy highly structured proper subclasses of the regular relations

and functions (Heinz, 2010a; Chandlee et al., 2018). In particular, many alternations can be modeled as *subsequential* functions (Heinz and Lai, 2013; Chandlee et al., 2014; Chandlee, 2017; Chandlee et al., 2018), and many local alternations fall within the class of k-*Input Strictly Local* (k-ISL) functions (Chandlee, 2014; Chandlee and Heinz, 2018). The important property of k-ISL functions is a locality condition bounded by some finite k.

For (iv), adopting a formally restricted hypothesis space has substantial typological and computational implications. In particular, it makes explicit what is expected to count as a possible phonological process, i.e., a mapping that falls within the subregular hypothesis (or within an explicitly motivated extension of it). This restriction in computational complexity also supports the maintenance of formal guarantees for learners targeting these classes, since learnability results can be stated relative to a precisely characterized hypothesis space.

For (v), we adopt inductive biases grounded in simplicity and locality. We take identity as the default hypothesis and treat learning as the task of identifying deviations from it. Concretely, the learner begins with the smallest plausible hypothesis, and expands it only when forced to do so by the data, and only in tightly constrained, local ways (similarly to the Belth (2023) D2L learner). These foundations set up the central question of the dissertation: *How can we learn phonological mappings from small data in a way that is both empirically effective and theoretically explanatory?*

1.1 Approaches to learning phonology

The dissertation is situated at the intersection of several research traditions that study phonological learning. These traditions differ not only in modeling choices, but in what they treat as the primary scientific object: rule systems, constraint rankings, neural architectures, or formally characterized classes of computations.

1.1.1 Rule-based learners and locality-biased rule induction

One long-standing tradition treats phonological grammars as *rule systems* and asks how such rules could be learned from alternations present in the lexicon. This includes early discovery procedures for ordered SPE-style rules (Johnson, 1984) and later models that infer and generalize rules from paradigmatic alternations, such as present–past pairs (Albright and Hayes, 2002). More recent proposals incorporate explicit simplicity principles, such as Minimum Description Length (Rasin et al., 2021). These approaches often align well with traditional phonological representations and can be empirically persuasive, for example by inducing plausible rules. However, they are typically not formulated around a sharply delimited computational hypothesis class, so it can be difficult to state general typological predictions or learning guarantees beyond performance on particular datasets.

Another representative example is the Parsimonious Local Phonology (PLP) learner from Belth (2023), PLP is designed to model a bias toward *local* generalizations. It learns grammars formalized as one or more phonological rules, and it incorporates phonological representations such as features and natural classes. PLP performs well in a variety of learning experiments and provides a clear computational story of how locality can be a built-in preference in the learner.

At the same time, rule-learning approaches (including PLP) typically prioritize empirical performance and psychological plausibility over formal characterization of the hypothesis space. In particular, it is often difficult to say precisely what class of mappings the learner can represent or is guaranteed to identify.

1.1.2 Constraint-based learning in Optimality Theory

In the constraint-based tradition, phonological grammars are modeled not as ordered rules but as rankings of constraints. In Optimality Theory (OT), grammars are defined by a constraint set CON, a candidate generator GEN, and an evaluation procedure.

The main idea is that different languages correspond to different rankings of the same universal constraints (Prince and Smolensky, 1993). This architecture changes what learning looks like such that learning becomes the task of inferring a ranking that selects attested winners over competitors.

A core algorithm in this tradition is Recursive Constraint Demotion (RCD), which learns a stratified hierarchy by repeatedly identifying constraints that must be ranked high given winner–loser comparisons. RCD is attractive from the perspective of formal learning theory because it comes with theoretical guarantees. Particularly under appropriate assumptions about the data (and candidate comparisons), it will converge to a ranking consistent with the evidence (Tesar and Smolensky, 1998, 2000; Prince and Tesar, 2004). Related approaches, including gradual or probabilistic updates of constraint strengths, have been developed to model variation and gradient behavior (e.g. (Boersma and Hayes, 2001)).

However, the OT learning picture illustrates a different sense in which "what class is being learned" can be hard to characterize. For a fixed CON, the space of possible grammars is the set of all rankings of CON. The typology induced by that space is called the *factorial typology*, which is the set of distinct patterns that can arise from all possible rankings of the given constraints. In principle, with n constraints there are $n!$ total rankings, which suggests that the overall typology can still be extremely large. Importantly, even if one restricts the *form* of constraints in significant ways, the resulting factorial typology can remain comparatively unconstrained in what patterns it includes (Lamont, 2021, 2022).

1.1.3 Neural network approaches and representation learning

Neural network models constitute a third major tradition in phonological learning. These models have produced a large and growing body of experimental results, including demonstrations that neural systems can learn nontrivial phonological generalizations from distributional patterns and can model certain aspects of productivity and alternation (Beguš, 2020; Prickett, 2021).

At the same time, neural approaches raise scientific questions that align closely with the themes of this dissertation. One concerns *representations*: it is not clear whether neural learners encode any phonological representation in a way that supports generalization, and even if they do, it is problematic to diagnose it due to their lack of interpretability (Rudin, 2019). A second concerns *expressive capacity versus typology*. Even if a network can fit training data, what does it predict about the complexity space of learnable processes, and how does that space relate to what generalizations are attested vs. unattested in natural languages? Finally, neural networks are often trained with enormous datasets, extensive hyperparameter tuning, and optimization procedures which does not in any way relate to human-learning conditions.

1.1.4 Grammatical inference and subregular phonology

The grammatical inference tradition develops learning algorithms whose starting point is to understand the conditions under which a learner can generalize appropriately from data (de la Higuera, 2010; Heinz et al., 2015; Heinz and Sempere, 2016; Wieczorek, 2017). It asks questions like: "What algorithm A (if any), when given certain data D exemplifying a process P drawn from a certain class C, will invariably return a grammar for P?" Despite offering formal guarantees, grammatical inference learners introduce their own practical challenges.

Even when a class is learnable in principle, the finite samples required for formal guarantees can be much larger than, and qualitatively unlike, the data available in

realistic settings (Gildea and Jurafsky, 1996). Such a sample is referred to as a *characteristic sample*: a finite set of examples sufficient to guarantee the learner's success. For phonological mappings, this means that the sample must include the relevant alternations and conditioning environments needed to identify the grammar that generates them. This highly specific sample requirement is important from the perspective of acquisition modeling, because children generalize from limited, uneven evidence (Yang, 2006). Ideally, artificial learners should therefore have some basis in the conditions of natural language acquisition. It also matters for computational work on low-resource languages, where sparse datasets are the norm and data-efficiency is essential (Wiemerslage et al., 2022; Nag et al., 2024).

1.1.5 Comparison of the approaches in a nutshell

	Rule-based	**OT**	**Neural**	**GI**
Uses phonological representations	✓	✓	✗	✗
Performs well on learning experiments	✓	✓	✓	✗
Theorems about learnable class	✗	✗	✗	✓
Theorems about learning behavior	✗	✓	✗	✓

TABLE 1.1: Learning criteria and their inclusion in different traditions of learning phonological processes.

The overall advantages and disadvantages of the approaches discussed above are summarized in Table 1.1. This summary should not be taken too literally: whether a particular approach satisfies a given criterion is not as black-and-white as the table suggests. In practice, many of these properties fall along a continuum, and the cells in the table would more accurately be represented by various shades of gray. Nonetheless, the table reflects our overall reading of the literature, and we believe the Yes/No entries capture the main strengths and weaknesses of each approach at a high level. A valuable research goal is to develop results that turn "No" cells into "Yes" cells for *any* of these traditions.

This dissertation pursues that goal for the final column, representing grammatical inference learners. In particular, we show how to incorporate phonological representations, including features and tiers, into grammatical inference learners for phonological mappings. We also introduce additional parameters that further improve learning in small-data settings. Finally, we develop a novel learner that preserves the core idea of subregular restriction while improving data efficiency and propose a conservative relaxation of prior methods for learning tier content. The next section provides a more detailed summary of these contributions.

1.2 Contributions

The key phonological representation exploited in this dissertation is *features*. The guiding intuition resembles a general principle in learning complex systems: even when a system is high-dimensional, its behavior can often be governed by *sparse dependencies* among a small subset of relevant variables. Valiant (2013, p. 20) illustrates this idea with a biological analogy: "the amount produced of our seventh protein may depend on the concentrations of three others—say, the third, twenty-first, and seventy-third." The point is that global complexity does not imply that every component depends on everything else. The update rather depends on a subset of such components. We argue that an analogous sparsity can be exploited in phonological learning when the mapping is expressed over features.

Accordingly, we adopt a framework that factors the learning problem by phonological feature. Under this strategy, each feature's output behavior is learned from the subset of information that actually conditions it. This factorization reduces the input alphabet, which in turn shrinks the size of the induced machines and reduces the amount of data required for reliable generalization.

The dissertation develops and further builds on this strategy in several steps, each of which contributes to the overarching goal of small-data learning with typological

clarity.

1.2.1 Factor by Feature (FxF)

We introduce the *Factor by Feature* (FxF) framework that allows for incorporating phonological features into grammatical inference learners. FxF decomposes a target mapping $f : \Sigma^* \rightarrow \Delta^*$ into a collection of feature-wise mappings, one for each feature dimension. Inference is then conducted over feature values rather than fully specified segments. FxF can in principle be paired with any learner for restricted phonological mapping classes. In this dissertation, we first instantiate it with SOSFIA (Jardine et al., 2014), a learner for learning structured subclasses of subsequential functions (including k-ISL) in linear time and data.

We empirically evaluate FxF on multiple alternations across four languages, showing that featural factorization can dramatically reduce the amount of data required for the desired generalization relative to segment-based learning, especially for processes that correspond to local changes that affect only a subset of features. In the best case scenarios, such as in learning vowel nasalization in English, the characteristic sample size was reduced by 98% (from 18,278 input-output pairs to avg. 188). In more complex cases involving interacting processes, such as final devoicing and ɔ-raising in Polish, FxF(SOSFIA) still produced a substantial reduction in characteristic sample size (from 17,206 to an average of 8,235, i.e., approximately 50%). At the same time, the residual sample requirements indicate that there is scope for further improvement in these settings.

1.2.2 Beyond factorization: learning k and choosing direction

Featural factorization also creates room for additional parameters that further improve efficient inference. In particular, we introduce (i) learning the locality bound k and (ii)

choosing the processing direction (left-to-right vs. right-to-left). Crucially, these parameters can be selected *per feature*, yielding a flexible learning architecture in which different aspects of the mapping may rely on different locality requirements and different directional dependencies.

We show that some features can be learned from very limited information. It particularly benefits features that undergo no alternation at all and which can be captured by the smallest possible hypothesis represented with a one-state machine. We further show that the direction in which an input string is processed interacts with locality and with the distribution of conditioning contexts. When the relevant conditioning information is encountered *before* the target under a particular processing direction, the learner can often avoid output withholding and can default more safely to identity on unattested regions. In this way, aligning processing direction with the location of conditioning information can effectively reduce what must be learned.

Once again, the benefits of incorporating these parameters are most pronounced for single-process, feature-changing mappings. For English vowel nasalization, for instance, while feature factorization reduces the average characteristic sample size to approximately 188 input–output pairs, changing the processing direction from left-to-right to right-to-left yields a further reduction to roughly 137 pairs. Allowing the learner to select the locality bound k independently for each feature produces an even larger gain, lowering the average sample size to about 44 pairs. By contrast, for processes such as ə-epenthesis in Chukchi, reducing k on a per-feature basis is not possible due to the nature of the process. Specifically, because epenthesis affects *all* features and the locality bound must remain the same, here $k = 3$, throughout. Although direction choice can still substantially affect individual feature learners, for example, learning the the ə-epenthesis for feature [labial] in Chukchi requires approximately 313 pairs under left-to-right processing but only about 152 under right-to-left, the overall sample size at the segmental level does not decrease significantly, indicating more limited benefits for this particular process.

1.2.3 A new learner: SEAL and identity-first output inference

While the FxF framework applied to SOSFIA, together with the two additional parameters introduced above, performs well on synthetically generated data, the same is not true for sparse datasets drawn from phonology textbooks. This motivates a new learner, SEAL, a deterministic transducer learning algorithm based on *string equations*. Given a choice of learning parameters (including k, feature projections, and processing direction), SEAL first constructs a k-ISL transducer *skeleton* and treats the output on each transition as an unknown variable. The training inputs are then run through the skeleton to produce *encoded inputs*, which represent the sequence of transition variables traversed by each example. Learning then is reduced to solving a system of equations that relates these encoded inputs to the observed outputs by assigning concrete substrings to the individual encoded input so that the assigned distribution is consistent across the sample.

Crucially, SEAL is guided by an *identity-first* inductive bias. It begins with the simplest hypothesis, a conservative identity-based alignment, and redistributes output material only when forced to do so by global inconsistency in the system of equations. This contrasts with locally greedy output-assignment strategies (e.g. methods driven by longest-common-prefix heuristics) and yields substantial gains in data efficiency in our experiments.

FxF(SEAL), relative to FxF(SOSFIA), yields substantial improvements in sample efficiency across all functions considered. The largest gains arise in settings where directionality and per-feature locality can be exploited most effectively. For example, in Yowlumne, in the best configuration settings, FxF(SOSFIA) requires approximately 164 input–output pairs on average, whereas FxF(SEAL) reduces this requirement to roughly 20 pairs. The advantage of SEAL persists even in more challenging cases. In Polish, average requirements decrease from 15,169 pairs under SOSFIA to 8,068 pairs in the best FxF(SOSFIA) configuration, but drop dramatically to approximately 104

input–output pairs under FxF(SEAL). Taken together, these results indicate that directionality and feature-specific locality *do* contribute meaningfully to sample-efficient learning, but that fully realizing these benefits may require a different approach to learning the output mapping, one for which SEAL appears to provide a promising direction.

1.2.4 From synthetic paradigms to textbook-style datasets

As noted above, one criticism of learnability results is that the characteristic samples that support formal guarantees may be unlikely to appear in realistic linguistic data (Gildea and Jurafsky, 1996). To address this concern, we evaluate SEAL on sparse, more naturalistic datasets adapted from phonology textbooks and problem sets compiled by Ellis et al. (2022). These datasets are sparse by design, involve larger and more diverse phonemic inventories than the synthetic experiments, and include interactions among multiple processes. In those experiments, we analyze not only consistency on the training pairs, but also generalization beyond the observed sample, including behavior on out-of-vocabulary forms and on unseen paths through the induced machines.

We test SEAL on five case studies drawn from Koasati, Kerewe, Lumasaaba, Samoan, and Serbo-Croatian. Across all five datasets, SEAL converges on a transducer consistent with the training data. Moreover, in four of the five cases, the learned transducer generalizes appropriately to unseen forms. The exception is Lumasaaba, where generalization appears limited, plausibly due to the very small dataset (14 datapoints), the complexity of the target mapping (involving four interacting phonological processes), and an inventory that is likely underspecified as a consequence of sparse sampling, which in turn constrains the potential benefits of feature-based generalization via factorization. Finally, because the learned hypotheses are *fully interpretable*, they allow

detailed error analysis and make it possible to isolate specific failure modes, providing clear targets for refinement in future work.

1.2.5 Tiers and long-distance processes

Not all phonological dependencies are local. Canonical cases include vowel harmony and consonant harmony (Clements and Sezer, 1982; Hansson, 2001), where trigger and target can be separated by arbitrarily many interveners. A standard phonological strategy is to introduce *tiers* that project only the relevant segments, rendering long-distance dependencies local on the projected tier, similarly to what has been done for tone (Goldsmith, 1976).

Building on prior work on tier-based strictly local languages and functions (Jardine and Heinz, 2016; Burness and McMullin, 2020; Burness et al., 2021), we develop a procedure for learning tier *content* from the inputs, along with a conservative relaxed variant designed specifically for small-data settings. This choice is also theoretically motivated by Lambert and Heinz (2024), who show that certain long-distance processes that have often been analyzed using output-oriented function classes can be reinterpreted in input-oriented terms. Under this reinterpretation, the relevant tier can be defined over the input, bringing these processes into the scope of the present learning framework.

To evaluate the tier inference component on naturalistic data, we apply it to a Turkish dataset in which vowels participate in both backness and rounding harmony. In both cases, the procedure identifies the appropriate tier content for the relevant features, yielding phonologically well-formed tiers that isolate precisely the elements relevant to each long-distance dependency.

Given an inferred tier (and imposing a memory bound of $k = 2$, since most long-distance processes do not require larger bounds), we then construct the corresponding canonical transducer skeleton for the algebraic *tier-based definite* class. Importantly, the

definite class is equivalent to the k-ISL class, which allows us to use SEAL to infer the transition outputs of this skeleton.

1.3 Outline

Chapter 2 introduces the formal background: deterministic finite-state transducers and the relevant subregular function classes, with English vowel nasalization as a running example, and a feature system following Hayes (2009). Chapter 3 introduces the FxF framework, explains why featural decomposition can reduce representation size and characteristic samples, and evaluates FxF (with SOSFIA) on processes from English, Yowlumne, Chukchi and Polish. Chapter 4 develops per-feature learning of k and examines the role of processing directionality. Chapter 5 introduces SEAL and its variants for handling alignment issues raised by deletion and epenthesis, and compares its empirical behavior to prior learning strategies. Chapter 6 evaluates SEAL on sparse, textbook-style datasets across five languages and provides a detailed analysis of what has and has not been learned. Chapter 7 extends the representational approach to tiers, developing an input-oriented tier learner and illustrating how SEAL can be used to infer deterministic transducers over learned tiers. Chapter 8 discusses limitations and future directions, including improving behavior on unseen paths and enriching representations where necessary, and concludes the thesis' contributions.

Chapter 2

Related Work and Preliminaries

2.1 Importance of Phonological Features

Phonological features have played a central role in linguistic theory, tracing back to the seminal work of Roman Jakobson and Nikolai Trubetzkoy (Jakobson et al., 1951; Trubetzkoy, 1969) and extensively developed in the subsequent theoretical framework, such as SPE (Chomsky and Halle, 1986). Features group speech sounds into natural classes, which both simplifies the description of phonological rules and enables generalization across typologically diverse languages (Clements and Hume, 1995). For example, a process which targets *voiced obstruents* can be characterized as a natural class with the representation ($\left[\begin{smallmatrix}+\\-\end{smallmatrix}\ \begin{smallmatrix}\text{voice}\\\text{sonorant}\end{smallmatrix}\right]$) rather than listing individual segments, such as [b d g v z ʒ d͡ʒ ɣ], within that class. Furthermore, features also make it possible to represent phonemic distinctions at various levels of representations (Kaye, 1980; Rubach, 2019). For example, in English, the distinction between aspirated voiceless stops is merely a surface one, while for native Thai speakers aspiration is a contrastive feature to distinguish the types of voiceless obstruents in the mental representation (Hyman, 1975, p. 26–27).

Beyond their role in stating phonological rules, there is independent evidence that features are needed to characterize the organization of phonological systems themselves. Typological studies of segmental inventories argue that cross-linguistic generalizations are best captured in terms of a set of distinctive features and some principles

aimed at organizing them (e.g., feature economy, robustness, phonological enhancement), rather than in terms of segments (Avery and Idsardi, 2001; Clements, 2009). Complementary work in contrastive hierarchy tradition argues that only contrastive features, which are arranged in a language-specific hierarchy, participate in phonological computation. Within this framework, such hierarchies are necessary to account for patterns such as neutralization or harmony in an empirically adequate and logically coherent way (Dresher, 2009, p. 11–35). In this view, it is not just the mere existence of features, but also their contrastive status, that is central to the structure of phonological grammars.

The importance of a featural level of representation is also supported by psycholinguistic evidence (Du and Durvasula, 2025) and speech-error studies (Fromkin, 1971), as well as by work on segment inventories, phonological alternations, and phonetic implementation (Cohn, 2011). These pieces of evidence support the idea that feature-based representations play an important role in phonology, independently of any particular theory of learning. At the same time, the nature of features remains a subject of debate (Duanmu, 2016). Traditional generative approaches argue that features are innate and are part of the learner's prior knowledge, on the grounds that some representational primitives are a logical precondition for any learning (e.g. Chomsky and Halle 1965, 1986; Fodor 1980; Reiss and Volenec 2022). We adopt this point of view here, while acknowledging that some researchers aim to infer features and natural classes from distributional patterns in the input (Mielke, 2008; Mayer, 2020). In practice, many computational models assume, as we do, that a feature system is 'baked' into phonological models (Gildea and Jurafsky, 1996; Hayes and Wilson, 2008).

Because phonological features have so much independent support, employing featural representations to learn phonological processes has ample precedents within the field of phonology. For example, Albright and Hayes (2002, 2003) provided a learning model which represents segmental changes with those sets of features that are the smallest natural class consistent with them. More recently, Belth (2023, 2024) present

a rule-learning model which incorporates the Tolerance Principle (Yang, 2016) and shows that it can generalize less conservatively. Despite the merits of these learning models and the simulations conducted with them, they do not have theoretical results with respect to their general learning behavior.

This paper adopts an alternative strategy: it begins with learning algorithms in the grammatical inference tradition, which come with some theoretical guarantees, and adapts them to incorporate features. In this way, this work bridges two communities: for the phonological community, it helps introduce them to results in grammatical inference, and for the grammatical inference community, it shows how to incorporate sub-symbolic structure into their algorithms.

Strother-Garcia et al. (2016) and Vu et al. (2018) show how model-theoretic approaches to formal languages allow one to naturally provide grammars which make reference to phonological features. Building on this work, Chandlee et al. (2019) and Rawski (2021) introduce the Bottom-Up Factor Inference Algorithm (BUFIA) which provably learns formal languages describable as the conjunctions of constraints, where the constraints are defined over arbitrary, but given, model-theoretic signatures. Recently, BUFIA has been shown to be practically effective for learning both phonotactic constraints (Swanson et al. 2025) and tonotactic constraints (Li 2025).

The line of research in this paper differs from this earlier research in two ways. First, it targets phonological processes (transformations of strings to strings) as opposed to phonotactics (well-formedness of strings). Second, the aforementioned works used logical representations of grammars, whereas this study uses finite-state representations of grammars. Related work by Yolyan (2025a,b) incorporates the ideas underlying BUFIA into a learning model for phonological processes described with the logical formalism known as Boolean Monadic Recursive Schemes (Bhaskar et al., 2020; Chandlee and Jardine, 2021). A detailed comparison to that approach and the one adopted here is left for future research.

2.2 Subregular Classes in Phonology

This work stems from the idea that phonology is computationally simple. Heinz (2011a,b, 2018, 2025) provide an overview and arguments for this idea. On the stringset side (phonotactics), Heinz (2010a) argues that local phonotactics belong to the Strictly Local class and that long-distance phonotactic patterns belong to the Strictly Piecewise class. As such, well-formedness depends only on contiguous substrings or non-contiguous subsequences. Subsequent logic-based characterizations (Rogers and Pullum, 2011; Rogers et al., 2013) and the consideration of the Tier-Based Strictly Local class (Heinz et al., 2011; McMullin, 2016; Lambert and Rogers, 2020; Lambert, 2023) further reinforce that phonotactic patterns occupy a very restricted complexity range. Furthermore, these classes, when parameterized, can be learned by simple mechanisms (Heinz, 2010b; Heinz et al., 2012; Jardine and Heinz, 2016; Jardine and McMullin, 2017; Lambert et al., 2021). Additional related classes for phonotactics are studied by Lambert (2026).

Of particular relevance to this study are subregular classes of string-to-string functions that can model various phonological processes. Long established results in computational linguistics show that phonological rules can be represented with subsequential functions and modeled with deterministic finite-state transducers (Kaplan and Kay, 1994; Sakarovitch, 2009). Chandlee (2014) and Chandlee and Heinz (2018) demonstrate that many phonological processes fall into small subclasses of subsequential functions: Input Strictly Local (ISL) and Output Strictly Local (OSL). The outputs in ISL functions are determined based on a bounded window of the *input* context, while in OSL functions the outputs depend on a bounded window of the *output* context. More complex processes might require projection of relevant elements onto tiers or other formal devices (Jardine, 2016; McCollum et al., 2020; Burness et al., 2021; Lambert, 2022; Lambert and Heinz, 2023, 2024).

The subregular classification of phonological mappings offers significant advantages for learnability. When a learner is equipped with the prior knowledge that the target function lies within a well-defined subregular class, the hypothesis space can be constrained accordingly, which can enable efficient generalizations. Such restrictions not only reduce the risk of overfitting but also provide formal guarantees of convergence and correctness. In the next subsection, we provide a review of algorithms that target distinct subregular subclasses of subsequential functions.

2.3 Subsequential function classes learners and their limitations

Oncina et al. (1993) introduce OSTIA, which successfully identifies any total subsequential function in the limit, given a positive characteristic sample. OSTIA enforces *onwardness*, a property which guarantees that outputs are produced as soon as they are determined. It constructs a prefix tree transducer and merges compatible states, which allows for further generalization to unseen data. While the algorithm runs in polynomial time, its cubic time and data complexity can be computationally expensive in practice (Gildea and Jurafsky, 1996).

Chandlee et al. (2014) present the ISLFLA algorithm that particularly learns the class of k-ISL functions. The learner, like OSTIA, builds a prefix tree transducer and merges states to produce a transducer with the appropriate structure. The counterpart for k-OSL functions was developed by Chandlee et al. (2015) and is called OSLFLA. It generalizes by constructing a finite-state transducer and iteratively exploring the states reachable from an initial state and associating transitions with the longest common prefix of the outputs observed in the training data. Both learners run in quadratic time and data. Thus, in practice, if one knows that the target process belongs to the k-ISL or k-OSL functions, then efficiency considerations would favor the use of ISLFA and

OSLFLA over OSTIA.

Jardine et al. (2014) introduce SOSFIA, which identifies particular subclasses of subsequential functions in *linear* time and data. SOSFIA assumes that the underlying structure of the transducer modeling the target function is known in advance. For many subregular classes, such as k-ISL functions, this is, in fact, the case: the class determines the structure of the transducer. Consequently, the learning problem is then reduced to calculating the outputs for each transition. Due to its efficiency relative to other learners, we adopt SOSFIA as the basis for illustrating the Factor-by-Feature approach proposed in this dissertation. Further technical details on the algorithm are provided in Section 3.3.

The learners discussed above show how the structure of the hypothesis can facilitate successful generalization. As Heinz (2010a, p. 636) argues, whether a grammar employs featural representations or not is orthogonal to other formal properties of the grammar, such as whether it describes string to string mapping that is subsequential, ISL, or OSL. Thus, the focus of the above works was on the contributions such formal properties can make to learning. That said, the aforementioned learners all share a core limitation in practice. In particular, they appear to require large, highly specific characteristic samples. Gildea and Jurafsky (1996, p. 507) note that the characteristic sample of OSTIA "may require types of strings that are not found in the language for phonotactic or other reasons." For example, it may require a string containing a *ttt* sequence. The practical consequence is that natural data sets do not contain characteristic samples. For learners such as OSTIA, which construct transducers from the data where each encountered data point has its representative path, missing data points lead to failure in generalization or incomplete functions. By contrast, ISLFLA detects when the sample is not characteristic and halts without returning a transducer. SOSFIA similarly refuses to output a transducer in the absence of a characteristic sample, though heuristics can be added to ensure that it outputs something.

Gildea and Jurafsky (1996) introduce phonologically motivated heuristics, including feature-based similarity biases and alignment assumptions that aim to improve the performance of OSTIA under sparse settings. Nevertheless, these modifications do not eliminate the dependence on highly specific characteristic samples required for successful identification. This limitation motivates further work on learning architectures that are less sensitive to the availability of such samples. One such approach, developed in this dissertation, is to incorporate phonological representations directly into the learning process.

Learners operating over symbols representing segments may not generalize across the segments that share some phonological features specifications. Put differently, symbols such as [b], [d], [g] are seen as entirely separate units, and their crucial similarity of being voiced obstruents is lost. This contrasts with phonological theory in which rules and constraints are described in terms of natural classes, and not segments in isolation. The FxF framework directly addresses this issue by shifting the learning problem from segment-level to feature-level processes. By identifying the target function in terms of each individual feature, FxF allows for generalization across any segment sharing particular feature values, which in turn reduces the data and improves interpretability. In this way, it directly synthesizes representations motivated by phonological theory with grammatical inference algorithms that come with theoretical guarantees.

2.4 Preliminaries

Let Σ denote a finite input alphabet of symbols and Σ^* the set of all strings over Σ of finite length. Such strings are called Σ-strings. The alphabetic symbols can represent either phonological segments, phonological feature values, or anything else that can be sequentially ordered. If w is a string, then w_i indicates the ith symbol in w and $|w|$ signifies the length of the string. λ is the empty string and $|\lambda| = 0$. For nonempty

strings w, it follows that $w = w_1 w_2 \dots w_n$ with $n = |w|$. The *prefixes* of w are defined as follows: $\text{Pref}(w) = \{u \in \Sigma^* : uv = w, v \in \Sigma^*\}$. Similarly, the *suffixes* of w is $\text{Suff}(w) = \{v \in \Sigma^* : uv = w, u \in \Sigma^*\}$. Also, $\text{Suff}(w)^{k-1}$ ($\text{Pref}(w)^{k-1}$) is a suffix (prefix) of length $k-1$ for some $k \in \mathbb{N}$.

Given Σ-strings u and v, their concatenation is denoted as uv. If $w = uv$, $u^{-1}w = v$. Given two alphabets Σ, Δ, a Σ-string u, and Δ-string v of the same length, their direct product is a $\Sigma \times \Delta$-string $u \times v = (u_1, v_1)(u_2, v_2) \dots (u_n, v_m)$.

Definition 2.1. *The longest common prefix (lcp) of a set of strings S, where $S \subseteq \Sigma^*$, is defined as follows:*

$$lcp(S) = x \text{ iff } x \in \bigcap_{w \in S} Pref(w), \forall x'(x' \in \bigcap_{w \in S} Pref(w) \Longrightarrow |x'| \leq |x|)$$

Let Δ denote an output alphabet. Given a relation $t : \Sigma^* \times \Delta^*$, for all $w \in \Sigma^*$ and for all $v \in \Delta^*$, if $(w, v), (w, v') \in t$ implies $v = v'$, then t is a *function* and we write $t(w) = v$. A function is *total* iff for all $w \in \Sigma^*$ there exists $v \in \Delta^*$ st. $t(w) = v$.

For any function $f : \Sigma^* \to \Delta^*$ and $w \in \Sigma^*$ we define the *tails* of x with respect to f as $\text{tails}_f(x) = \{(y, v) : f(xy) = uv, u = \text{lcp}(f(x\Sigma^*))\}$. Informally, the tails of x with respect to f is the function identified with the input/output pairs (y, v) that extend from the input x and the lcp of all of possible outputs from input strings beginning with x $(x, lcp(f(x\Sigma^*))$. Two strings x and x' are *tail-equivalent* with respect to a function f iff $\text{tails}_f(x) = \text{tails}_f(x')$. We also denote it as follows: $x \sim_f x'$, where $\sim_f$ is the equivalence relation induced by tails_f which partitions Σ^*. A *subsequential function* is a function f that $\sim_f \Sigma^*$ into finitely many blocks (Choffrut, 1977, 2003; Oncina and Garcia, 1991).

2.4.1 Subsequential Finite-State Transducer

There are different ways to represent finite-state transducers that define subsequential functions (Oncina et al., 1993; Mohri, 1997; Choffrut, 2003). We use the delimited

onward finite-state transducer (Jardine et al., 2014) because it associates every output with a transition, which simplifies presentation and analysis of the SOSFIA learning algorithm. Formally, onwardness relates the states in the transducer to the blocks induced by the tails equivalence relation. Informally, it ensures that outputs are produced as early as possible.

Definition 2.2. *A delimited onward subsequential finite-state transducer (DFT) is a tuple* $(Q, \Sigma, \Delta, q_0, q_f, \delta)$ *where* Q *is a finite set of states;* Σ *and* Δ *are input and output alphabets, respectively;* $q_0 \in Q$ *is the* initial *state;* $q_f \in Q$ *is the* final state*; the transition relation is* $\delta \in Q \times (\Sigma \cup \{\rtimes, \ltimes\}) \times \Delta^* \times Q$*;* $\rtimes$ *and* $\ltimes$ *are the left and right boundary symbols respectively, and the following holds:*

i. *if* $(q, a, v, r) \in \delta$ *then* $q \neq q_f$ *and* $r \neq q_0$ *(there are no outgoing transitions from the final state nor any transitions incoming to the initial state),*

ii. *if* $(q, a, v, q_f) \in \delta$ *then* $a = \ltimes$ *and* $q \neq q_0$ *(all transitions to the final state have as input the right word boundary and do not originate in the initial state),*

iii. $(q_0, a, v, r) \in \delta$ *then* $a = \rtimes$ *(all transitions out of the initial state have as input the left word boundary)*

iv. *if* $(q, \rtimes, v, r) \in \delta$ *then* $q = q_0$ *(all transitions which have as input the left word boundary originate in the initial state),*

v. *if* $((q, a, v, r), (q, a, u, s)) \in \delta$ *then* $(v = u)$ *and* $(r = s)$ *(determinism),*

vi. $(\forall q \in (Q \setminus q_0))[lcp\{v \in \Sigma^* | (\exists a \in \Sigma, r \in Q)[(q, a, v, r) \in \delta]\} = \lambda]$ *(onwardness).*

The function a DFT τ *recognizes can be defined by establishing a recurrence between input strings, states, and output strings. Informally, the idea is that given a state* q *and with an output string* y *already written, we can process an input string* x *letter by letter until its end, and return an updated output string* y'*. Formally, this function has type* $\pi : Q \times \Delta^* \times \Sigma^* \to \Delta^*$

and is defined as follows.

$$\begin{aligned}\pi(q, y, \lambda) &= y \\ \pi(q, y, ax) &= \pi(q', yv, x) \textit{ provided } (q, a, v, q') \in \delta\end{aligned}$$

Finally, the function defined by τ *is* $\{(x, y) \in \Sigma^* \times \Delta^* \mid \pi(q_0, \lambda, \rtimes x \ltimes) = y\}$. *We write* $\tau(x) = y$ *iff* $(x, y) \in \tau$.

For additional details regarding these DFTs see Jardine et al. (2014).

2.4.2 Input Strictly Local (ISL) Functions

One class of subsequential functions that we focus on in this paper are Input Strictly k -Local ($k-$ISL) functions defined below in Definition 2.3 (Chandlee et al., 2014).

Definition 2.3. *A function* f *is ISL iff there is a* $k \in \mathbb{N}$ *such that for all* $u_1, u_2 \in \Sigma^*$ *if* $Suff^{k-1}(u_1) = Suff^{k-1}(u_2)$ *then* $tails_f(u_1) = tails_f(u_2)$. *A function is* k*-ISL if it is ISL for some* k.

We limit our attention to total k-ISL functions. Informally, for every total ISL function f, there is a k marking the length of the widest substring required to define f, and there is a DFT representing f which is structured in a particular way: the current state of the machine is always determined by the previous $k-1$ symbols read in the input. Such DFTs are called k-ISL DFTs. Chandlee et al. (2014) provide an automata-theoretic characterization of ISL functions, Lambert and Heinz (2023) characterize them algebraically, and their relevance to phonology is discussed in Chandlee and Heinz (2018) and Chandlee et al. (2018).

Definition 2.4. *A total* k*-ISL DFT can be constructed as follows:*

- $Q = \Sigma^{\leq k-1} \cup \{q_0, q_f\}$,
- $\exists v \in \Delta^*$ *s. t.* $(q_0, \rtimes, v, \lambda) \in \delta$.

- $\forall q \in Q \backslash \{q_0, q_f\}, \exists v \in \Delta^* \text{ s. t. } (q, \ltimes, v, q_f) \in \delta.$
- $\forall q \in Q \backslash \{q_0, q_f\}, \forall a \in \Sigma, \exists v \in \Delta^* \text{ s. t. } (q, a, v, \mathit{Suff}^{k-1}(qa)) \in \delta.$

Note that this construction may not be the smallest onward DFT recognizing a k-ISL function. An important insight resulting from this construction of a k-ISL DFT is the dependency of $|Q|$ on the parameter k and the cardinality of Σ. In other words, the larger the k or the larger the alphabet Σ, the larger the state space. The cardinality of Q can be calculated as follows: $|Q| = \sum_{n=0}^{k-1} |\Sigma|^n + 2$.[1]

2.4.3 Example: English Vowel Nasalization

Let us consider a concrete example of a 2-ISL function, such as English vowel nasalization. In this process vowels are nasalized when followed by nasal consonants. Solé (1995), Chen (1997), and Proctor et al. (2013) show that nasalization is systematically conditioned and that listeners perceive nasalized vowels as distinct from their oral counterparts. Generative analyses of English phonology typically assume that vowels are underlyingly oral and undergo regressive nasalization when followed by tautosyllabic nasal consonants (Selkirk, 1972; Hayes, 2009). The traditional view, advocated by Krakow and Huffman (1989) and formalized by Krämer (2015), treats the alternation as a predictable, non-contrastive feature-spreading rule. We adopt this view in our study.[2] For the purpose of this study we do not introduce syllable boundaries to the data and we represent the process as follows:

(1) Vowel Nasalization (VN)

$$[-\text{cons}] \longrightarrow [+\text{nasal}] \;/\; \_\_ \left[\begin{smallmatrix} +\text{cons} \\ +\text{nasal} \end{smallmatrix} \right]$$

[1] The number 2 includes the initial state q_0 and the final state q_f.

[2] Alternative analyses exist. In Optimality Theory (Prince and Smolensky, 1993) under the principle of Richness of the Base (Kager, 1999), the underlying form may be considered already nasalized. Some studies have argued that nasalization arises from gestural overlap between the articulatory movements of a vowel and a following nasal consonant (Cohn, 1993; Krakow, 1993; Fowler and Brown, 2000), in which case, nasality can be interpreted as an automatic product of coarticulation and does not require reference to phonological processes per se.

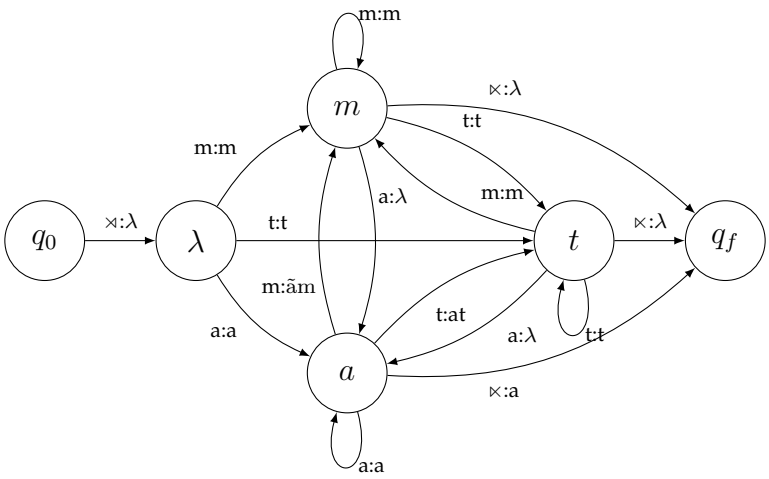


FIGURE 2.1: A 2-ISL DFT modelling vowel nasalization in English for $\Sigma = \{a, m, t\}$ and $\Delta = \Sigma \cup \{\tilde{a}\}$.

We now turn to explaining how this process can be modeled with a 2-ISL DFT, using a toy alphabet $\Sigma = \{a, m, t\}$. With this alphabet, the cardinality of Q is 6. A 2-ISL DFT representing a full inventory of English phonemes will quickly grow in size as every phoneme needs to be represented in a separate state. Figure 2.1 above presents a 2-ISL DFT for $|\Sigma| = 3$. For better readability, the transition from λ to the final state (q_f), the empty string transition, is omitted. The outputs on the transitions to and from the initial and final states are specified to be the empty string. The outputs on the transitions from the λ state to the m, a, and t states are m, λ, and t, respectively.

Consider how the DFT depicted in Figure 2.1 maps /mat/ to [mat] (as shown in Table 2.1) and /tam/ to [tãm] (as shown in Table 2.2).

Input:		⋊		m		a		t		⋉	
States:	q_0	→	λ	→	m	→	a	→	t	→	q_f
Output:		λ		m		λ		at		λ	

TABLE 2.1: How the DFT in Figure 2.1 maps /mat/ to [mat].

Input:		⋊		t		a		m		⋉	
States:	q_0	→	λ	→	t	→	a	→	m	→	q_f
Output:		λ		m		λ		ãm		λ	

TABLE 2.2: How the DFT in Figure 2.1 maps /tam/ to [tãm].

Because nasalization is conditioned by a right context, the transitions into the 'a' state from a distinct state outputs λ because at this point, it is unknown whether the following segment will be nasal or not so it is unknown whether to output 'a' or 'ã.' Once the segment following 'a' is read, the output produces either 'a,' 'at,' or 'ãm' appropriately. With the growth of the alphabet, it would be observed that all vowel states behave in a similar manner, a generalization that *can* be captured with featural but not segmental representations. The following section provides details on how we incorporate phonological features into the learning process.

2.4.4 Features

In this work, we consider a feature ϕ to be a function from an alphabet Σ to the set of values $V = \{+, -, 0\}$, which indicate positive, negative, or a null value for the feature respectively. As an example, consider the feature system for English in Table 2.3, adapted from Hayes (2009, p. 70 – 99), specifies several features for the phonemes of English.[3]

Each feature ϕ can be lifted to a homomorphism from Σ^* to V^* in the natural way. For instance, [cor](mat) $= [\,-\ 0\ +\,]$ and [cons](mat) $= [\,+\ -\ +\,]$. We call $\phi(x)$ the projection of x to the feature ϕ.

Given an ordered list of features $\Phi = (\phi_1, \phi_2, \ldots \phi_n)$, we interpret Φ as the direct product of the homomorphisms of its component parts. For instance, $[\,\substack{\text{cor}\\\text{cons}}\,](mat) = \left[\begin{smallmatrix} - & 0 & + \\ + & - & + \end{smallmatrix}\right]$. As is common in the phonological literature, we write these strings of feature sets with square brackets instead of the more cumbersome $(-,+)(0,-)(+,+)$. Like before, we say $\Phi(x)$ is the projection of x to the features in Φ. Note that the domain of Φ is Σ^* and its co-domain is $(V^n)^*$.

[3]Because vowels require fewer features to be fully specified, any feature that Hayes (2009) does not use for vowels is assigned the value 0 in our system, so the feature [coronal] for a vowel is represented as [0 coronal]. To keep the notation concise, we abbreviate feature names to their first few letters, so [cons] stands for [consonantal] and [cor] stands for [coronal].

Segment	æ	ɑ	ɔ	ɹ	ɛ	ʌ	ʊ	u	i	m	p	b	f	v	θ	ð	n	t	s	z	l	t͡ʃ	d͡ʒ	ŋ	k	g
high	-	-	-	+	-	-	+	+	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	+	+	+	+
low	+	+	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	+	-
tense	0	0	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0
front	+	-	-	+	+	-	-	-	+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0
back	-	+	+	-	-	+	+	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0
round	-	-	+	-	-	-	+	+	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0
long	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0
cons	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0
son	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
cont	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
delrel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
approx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nasal	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-
voice	+	+	+	+	+	+	+	+	+	+	-	-	-	+	-	+	+	-	+	+	+	+	+	+	+	-	+
lab	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
labdent	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cor	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ant	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
distr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
strid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
lat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
dor	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TABLE 2.3: Feature chart for English phonemes.

Finally, given a sample S of input/output pairs and two sets of features Φ, Φ', let $\langle \Phi, \Phi' \rangle(S) = \{(\Phi(x), \Phi'(y)) : (x, y) \in S\}$. In other words, for any pair of strings (x, y) we can project the x string to a set of feature values from Φ and the y string to a set of feature values from Φ'.

Given a feature system F, we observe that there are maximally $|V|^{|F|}$ distinct representations (see Reiss (2012) for discussion). This is the *maximal size of the featural alphabet*. Throughout the remainder of this paper, we use Σ_Φ to denote featural alphabet and Σ_{seg} to denote the segmental alphabet.

Chapter 3

FxF Framework and Preliminary Results

3.1 Decomposing functions with features

We aim to decompose the target subsequential function $f : \Sigma^* \rightarrow \Delta^*$ into n-many subsequential functions, one for each feature ϕ. In this work, each ϕ is individually represented with a k-ISL DFT, $\tau_{(\Phi,\phi)}$, where ϕ is the feature whose values are being output by the machine, and Φ is a set of features being used as input symbols by the machine. Put differently, each $\tau_{(\Phi,\phi)}$ can be thought of as aiming to predict a particular feature ϕ of the output using only the feature set Φ in the input. The feature combination Φ may vary across the n-many transducers.

For instance, consider the prior example of English vowel nasalization. Table 2.3 provides 22 features and so this process will be represented by 22 transducers, one for each feature. In this work, we assume each of these will be represented with a 2-ISL DFT, though we discuss in the conclusion why we believe this assumption can be relaxed in future work. All but one of these features, [nasal], is always faithful to its underlying representation. Consequently, for a faithful feature ϕ, the transducer $\tau_{(\{\phi\},\phi)}$ will be the 2-ISL DFT whose input and output alphabets are given by the value of set of ϕ, and which realizes the identity mapping over that set. Specifically, for each

transition, an input symbol not equal to $\rtimes$ or $\ltimes$ is an element of V and the output is an element of V^*. The outputs on the transitions with an input symbol $\sigma \in \{\rtimes, \ltimes\}$ is the empty string λ.

In English vowel nasalization, the feature [nasal] is not always faithful. Furthermore, its surface value cannot be predicted from the feature [nasal] alone. However, the features [nasal] and [consonantal] together are sufficient to determine its output value. Thus the feature [nasal] can be modeled with the 2-ISL DFT $\tau_{\left(\left[\begin{smallmatrix}\text{nas}\\ \text{cons}\end{smallmatrix}\right],\text{nasal}\right)}$. The input alphabet (and thus the states) of this DFT correspond to the set of feature values combinations of [nas, cons] in the input.

While there are 3^2 logical combinations available, in practice we only use those combinations that are present in a particular dataset. In the case of English, this means there are no [0 nasal] and [0 consonant] phonemes. Since both of those particular features have binary values $\{+, -\}$, this transducer could be composed, in theory, of up to 2^2 states, i.e. $\{\left[\begin{smallmatrix}+\\ +\end{smallmatrix}\right], \left[\begin{smallmatrix}-\\ +\end{smallmatrix}\right], \left[\begin{smallmatrix}-\\ -\end{smallmatrix}\right], \left[\begin{smallmatrix}+\\ -\end{smallmatrix}\right]\}$, where the upper symbol in each pair represents values for feature [nasal], and the lower one represents values for [consonantal]. As such, the state $\left[\begin{smallmatrix}+\\ +\end{smallmatrix}\right]$ corresponds to a nasal consonant, $\left[\begin{smallmatrix}-\\ +\end{smallmatrix}\right]$ corresponds to an oral consonant, and so on. However, in this analysis of English vowel nasalization, there are no underlying nasal vowels; hence the state $\left[\begin{smallmatrix}+\\ -\end{smallmatrix}\right]$ can also be omitted from this transducer.

Figure 3.1 represents $\tau_{\left(\left[\begin{smallmatrix}\text{nas}\\ \text{cons}\end{smallmatrix}\right],\text{nasal}\right)}$ for vowel nasalization in English with the initial, final and lambda states omitted for better readability. Also, note the difference between the input and output notations: the square brackets around the inputs indicate a collection of values for a single segment. The outputs are strings of values for the feature nasal only. For example, $\left[\begin{smallmatrix}+\\ +\end{smallmatrix}\right] : ++$ means that a $\left[\begin{smallmatrix}+ & \text{nas}\\ + & \text{cons}\end{smallmatrix}\right]$ symbol is being read and two [+nasal] symbols are being output.

This example may appear to suggest that the feature-based model of English nasalization is more complex than the segment-based model. The feature-based model consists of 22 DFTs and the segment-based model contains a single DFT. Furthermore,

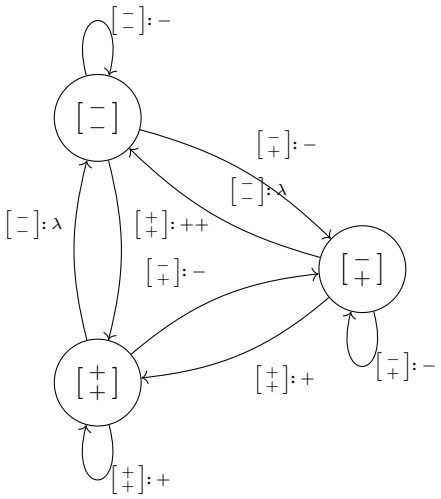


FIGURE 3.1: A 2-ISL DFT$_{([\substack{\text{nasal}\\\text{cons}}],\text{nasal})}$ for English vowel nasalization.

as each of the DFTs is a 2-ISL DFT with $|\Sigma_\Phi| \leq 3$, they roughly have the same number of states. So the total number of states in the feature-based model is roughly 22 times that of the segment-based model. However, it is important to notice that the segment-based DFT in Figure 2.1 is built only for 3 segments: {a, m, t}. The crucial difference between the transducer operating over segments from the model operating over features is that as the size of the segmental inventory grows, the number of states will increase in the segment-based model, but not in the feature-based model. This is the particular reason why learning subsequential functions over features requires less data. This point is discussed in greater detail in the next section.

Finally, we need to explain precisely how the feature-based model with its n-many DFTs (one for each $\phi \in F$) produces an output from an input like /tam/. Recall the notation $\tau_{(\Phi,\phi)}$ for each of these DFTs, where Φ is the set of features determining the input alphabet and ϕ is the feature whose values are being output. Essentially, an input string x is projected according to Φ and then processed by $\tau_{(\Phi,\phi)}$, producing a string of feature values for ϕ. This happens for each $\tau_{(\Phi,\phi)}$ in the feature-based model, and thus n-many strings of feature values are produced. The feature values across strings are then combined by the direct product operation in order to recover the segmental units. This workflow is schematized in Figure 3.2.

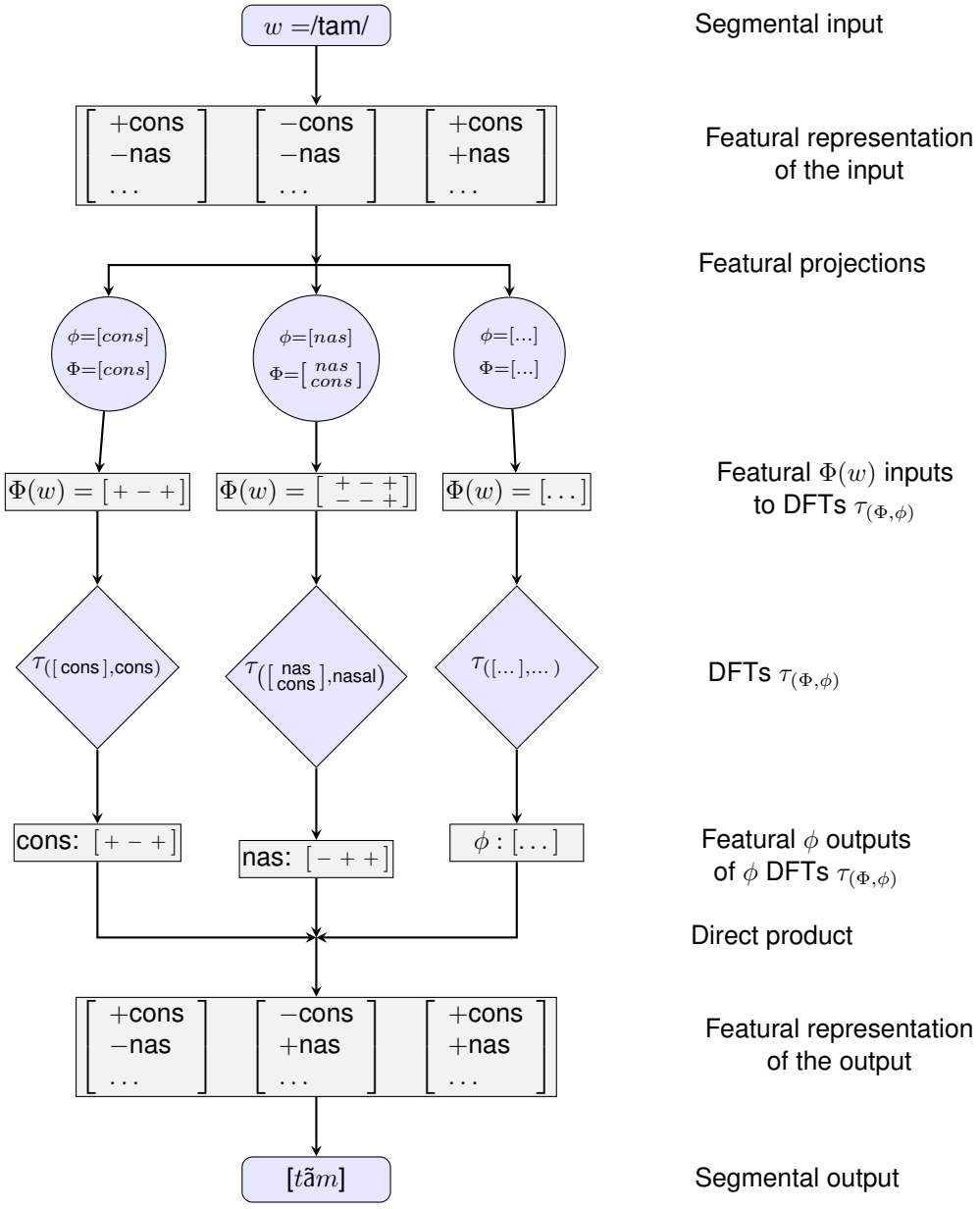


FIGURE 3.2: The generation process by which a feature-based model generates outputs from inputs.

3.2 Why featural decomposition should facilitate learning

This section argues on theoretical grounds that learning the feature-based representations will need less data. For concreteness, we make the argument with k-ISL DFT transducers.

Jardine et al. (2014) prove that the size of a characteristic sample for a target DFT τ is $O(|\tau|)$ (Lemma 18). In other words, the size of the characteristic sample is a linear function of the size of the transducer. The size of a DFT τ is measured by the number of its states and the lengths of its outputs on its transitions (Definition 3.1).

Definition 3.1. *The size of a DFT* $\tau = (Q, \Sigma, \Delta, q_0, q_f, \delta)$ *is* $|Q| + \sum_{(q,a,v,r)\in\delta} |v|$.

Following Definition 2.4, the number of states in a k-ISL DFT is simply the cardinality of $\Sigma^{\leq k-1} + 2$. Because the size of a k-ISL DFT transducer is dependent on $|\Sigma|$ and k, reducing one of these variables can reduce the size of the DFT. Table 3.1 presents an example of how the state space grows in terms of $|\Sigma|$ and k. As k increases, the number of states increases exponentially. For $k = 3$, if the alphabet consists of 5 segments, the number of states will grow to 33, while for $|\Sigma| = 10$, the number of states will increase to 113.

$\|\Sigma\|$	$k = 2$	$k = 3$	$k = 4$
5	8	33	158
10	13	113	1,113
25	28	153	3,278
50	53	2,553	127,553
100	103	10,103	1,010,103

TABLE 3.1: The number of states in k-ISL DFTs.

What do these numbers mean for segment-based models vis a vis feature-based models? In a segment-based model, Σ is the alphabet of segments. In a feature-based

model, Σ is the number of feature value combinations used in a particular feature machine.

Consider the example discussed earlier for English vowel nasalization. For concreteness, assume a variety of English with 40 phonemes. It follows that the segmental model would consist of a single 2-ISL DFT with 43 states. Now consider the featural model. This consists of 22 2-ISL DFTs. For faithful features like [consonantal], there are only 2 feature values $\{+, -\}$, which constitute the input alphabet. This [consonantal] DFT would have 5 states. For a vowel feature like [high], there are 3 feature values $\{+, -, 0\}$. Thus, the [high] DFT has 6 states. For the [nasal] DFT depicted in Figure 2.1, there are also 6 states, because only three combinations of the features [nasal, consonant] are present on the input side in any data sample (recall that the combination $[\begin{smallmatrix}+\\-\end{smallmatrix}]$ is omitted since the underlying vowels are not nasal).

As mentioned, the size of a characteristic sample to learn these DFTs is linear in the size of the DFT. Each of the feature DFTs is a fraction of the size of the segmental DFT, and it follows that the sample needed to generalize correctly is correspondingly a fraction of the size of the characteristic sample to learn the segmental DFT. In other words, a smaller machine results in a smaller characteristic sample. Section 3.4 provides empirical evidence supporting this hypothesis.

There are some caveats worth discussing, which helps motivate the subsequent empirical analysis. It is the case that the total size of the feature-based model is the sum of the sizes of the 22 DFTs. If each DFT had six states, the total size of the feature-based model would thus be 132, almost three times the size of the segmental model. It is natural to wonder whether the feature-based model is really a smaller model.

There are two reasons why the comparison in the above example can be misleading. First, this example is particularly simple in order to help explain the basic concepts. Processes with larger k windows require much larger DFTs; in fact, the number of states grows exponentially with respect to k. For instance, suppose there is a language with 50 phonemes described with 20 binary features (so each $\phi \in F$ has only 2

values) and consider a 3-ISL process where each feature value can be predicted from a combination of two features (so the $|\Sigma_\Phi| = 4$). A segment based model needs 2,553 states (Table 3.1). Each DFT in the feature model would have at most $20 + 3$ states, and thus the entire model contains $20 \times 23 = 460$ states. In other words, as the k value increases, feature-based models can be significantly more compact.

One potential objection to this line of reasoning is that as the number of feature combinations grow, the size of Σ in the feature-based model can grow as well. For example, if there was one feature value that required 10 binary features to predict that could potentially mean an alphabet of size 2^{10}. This objection is valid to a certain extent. We acknowledge that in the worst case, the feature-based model may be no better than the segmental based model, and could even be worse. However, we also do not anticipate that the worst case appears very often. For example, we are not aware of any phonological processes that required 10 features interacting to predict the value of a feature. Feature systems are also generally 'sparse' in the sense that one can usually find fewer than 10 features to uniquely pick out a phoneme. In other words, the specter of the worst case may be more theoretical than found in actual practice.

The second reason for comparing the total size of the segmental model to feature-based model is that the size of the characteristic sample for the feature-based model may not be the sum of the sizes of the characteristic samples of the individual DFTs that make it up. Those characteristic samples may overlap quite significantly.

In short, the questions raised by this theoretical analysis motivate the empirical investigation in the next section. We believe the results there vindicate the approach advocated for here.

3.3 The algorithms

This section describes three algorithms central to this study. First we discuss the SOSFIA algorithm introduced by Jardine et al. (2014). Next we present the Factor by Feature (FxF) algorithm which uses SOSFIA. Once again, we emphasize that the FxF algorithm is a broader concept that can work with *any* inference algorithm. Third, we present an algorithm that selects *a near-minimal sample* from the characteristic sample for SOSFIA. We call this algorithm A Minimal Sample Search Algorithm (AMßA).

3.3.1 SOSFIA

SOSFIA takes as its arguments a finite sample of input-output pairs $S \subset \Sigma^* \times \Delta^*$, and an output-empty DFT $\tau_\square$, i.e. a transducer where for every $(q, a, v, r) \in \delta$, $v = \square$, where $\square$ represents a 'blank' that needs to be filled in. Each output-empty DFT corresponds to a class of subsequential functions which vary only in what strings are assigned to the blanks. Many important subregular classes of functions can in fact be represented by such a DFT, such as the k-ISL functions. Jardine et al. (2014) prove that the algorithm successfully infers the class of subsequential functions represented by $\tau_\square$ given a sufficient data sample.

Algorithm 1: Structured Onward Subsequential Function Inference Algorithm (SOSFIA, Jardine et al. (2014))

Data: A sample $S \subset \rtimes\Sigma^* \ltimes \times\Delta^*$, an output-empty DFT $\tau_\square = (Q, \Sigma, \Delta = \{\square\}, q_0, q_f, \delta)$
Result: $\tau_\square$ as a DFT with filled transitions

```
F ← empty_Queue
Push (F, (q_0, λ))
mark (q_0)
while F ≠ ∅ do
    (q, w) ← Shift_First(F)
    for all σ ∈ Σ ∪ {⋊, ⋉} (in lexicographic order) do
        for all δ_i = (q, σ, □, q') ∈ δ do
            if ∃σ' ≠ σ such that (q, σ', u, q'') ∈ δ then
                Change δ_i to (q, σ, v, r), where v = min_change_S(w, σ)
            else
                Change δ_i to (q, σ, λ, r)
            end if
            if q' is not marked then
                Push (F, (q', wσ))
            end if
        end for
    end for
end while
```

SOSFIA conducts a *breadth-first* search through the output-empty DFT considering the shortest path that led to that state. In the case of the k-ISL DFTs, the shortest path is essentially encoded in the state label. For example, given a 3-ISL DFT, the shortest input prefix that leads to the state 'ma' state is '⋊ma'. SOSFIA stores in a queue untreated states together with their shortest prefixes. The next step identifies the outputs on the transitions using two functions `common_out` and `min_change`, defined below.

Definition 3.2. *The* `common_out` *of an input prefix w in a training sample S is the longest common prefix (lcp) of all the strings that start with w in S:*

$$\texttt{common\_out}_S(w) = \texttt{lcp}(\{u \in \Sigma^* : \exists v \ s.t. \ (wv, u) \in S\}).$$

Definition 3.3. *The* `min_change` *from a string w to a string $w\sigma$ in a training sample S:*

$$\texttt{min\_change}_S(\sigma, w) = \begin{cases} \texttt{common\_out}_S(\sigma) & \textit{if } w = \lambda \\ \texttt{common\_out}_S(w)^{-1}\texttt{common\_out}_S(w\sigma) & \textit{otherwise} \end{cases}$$

Given a transition $(q, a, \square, r)$, SOSFIA calculates the `lcp` of all the outputs associated with the input strings beginning with the shortest prefix to q, as well as the `lcp` of all the outputs associated with the input strings beginning with the shortest prefix to r by way of q and a. Then the minimum change between `common_out`(q) and `common_out`(r) is determined and replaces the blank square in the $(q, a, \square, r)$ transition. The procedure is repeated until the queue tracking untreated states is empty.

3.3.2 Factor by Features (FxF)

The `FxF` algorithm decomposes subsequential data into subsegmental units (features) and learns functions for those individual units, separately. Given a set of features F of cardinality n, a sample S (input-output pairs with segmental representations), a k value, the `FxF` algorithm iterates through the features $\phi \in F$ and for each ϕ it finds a set of features Φ such that the k-ISL DFT $\tau_{(\Phi,\phi)}$ accurately predicts those feature values in the outputs of the sample. The final output of `FxF` is a grammar G containing n-many $\tau_{(\Phi,\phi)}$, one for each $\phi \in F$. The pseudocode is presented in Algorithm 2.

Algorithm 2: Factor by Feature (FxF)

Data: A sample $S \subset \rtimes\Sigma^* \ltimes \times\Delta^*$, a feature set F, and a non-negative integer k
Result: A set G containing n-many k-ISL DFTs, one for each $\phi \in F$

```
G ← ∅
for all φ ∈ F (in lexicographic order) do
    τ_(Φ,φ) ← FeatSearch(k, S, F, φ, [{φ}])
    G ← G ∪ {τ_(Φ,φ)}
end for
return G
```

As discussed earlier, it is often the case that information from more than one feature is required to determine the output feature values of a feature ϕ. `FeatSearch` (Algorithm 3) searches for a featural combination which makes accurate predictions

for ϕ. It takes as arguments a non-negative integer k, the sample S, the feature system F, the target feature ϕ and a list of feature sets `FeatLIST`, which is initialized with one item in the list: the singleton feature set $\{\phi\}$.

The algorithm `FeatSearch` dequeues the first feature combination Φ from `FeatLIST` and evaluates it. It checks if the projected sample $S_{(\Phi,\phi)}$ is functional using the `is_functional` function (Definition 3.4). If it is, a new k-ISL DFT $\tau_{(\Phi,\phi)}$ is constructed. Definition 2.4 specifies the structure of a k-ISL DFT $\tau_{(\Phi,\phi)}$. At this step, we initialize an output-empty $\tau_{\square(\Phi,\phi)}$, i.e. one whose output on each transitions is empty. More formally, $\forall(q, a, v, r) \in \delta^*[v = \square]$. The newly constructed DFT and $S_{(\Phi,\phi)}$ are then passed as arguments to SOSFIA and `FeatSearch` returns the output of SOSFIA.

Definition 3.4. *A data sample S is functional under the projection functions Φ and Φ' iff for all $(u, v), (u', v') \in S$, if $\Phi(u) = \Phi'(u')$ then $\Phi(v) = \Phi'(v')$.*

Algorithm 3: `FeatSearch`

Data: A nonnegative integer k, a sample $S \subset \rtimes\Sigma^* \ltimes \times\Delta^*$, a feature set F, a feature ϕ, and a list of sets of features FeatLIST
Result: A DFT $\tau_{(\Phi,\phi)}$

```
while FeatLIST is not empty do
    Φ ← dequeue(FeatLIST)
    m ← |Φ|
    if is_functional(S_(Φ,φ)) then
        τ□ ← build output-empty k-ISL DFT with Φ and φ
        return SOSFIA(τ□, S_(Φ,φ))
    end if
end while
NewFeatLIST ← feat_combo(φ, m + 1, F)
return FeatSearch(k, S, F, φ, NewFeatLIST)
```

If $S_{(\Phi,\phi)}$ is not functional, on the other hand, `FeatSearch` generates a new list of feature combinations of size $m+1$ with the `feat_combo` function (Definition 3.5) and recursively calls itself with this new list appended to the rear of the queue, searching for a sufficient featural combination.

Definition 3.5. *`feat_combo`(ϕ, m, F) (Algorithm 4) produces a list of feature sets where each feature set contains ϕ and is of size m. Formally, `feat_combo`(ϕ, m, F) returns the set $\{\Phi \subseteq F \mid \phi \in \Phi, |\Phi| = m\}$ and puts its elements in a queue.*

Algorithm 4: `feat_combo`

Data: A feature ϕ, a positive integer m, and a feature set F
Result: A queue FeatLIST of feature sets Φ such that $\phi \in \Phi$ and $|\Phi| = m$

```
FeatLIST ← empty queue
AllCombos ← COMBINATIONS(F, m)
for all Φ ∈ AllCombos do
    if φ ∈ Φ then
        enqueue(FeatLIST, Φ)
    end if
end for
return FeatLIST
```

Algorithm `feat_combo` generates candidate feature sets by enumerating `COMBINATIONS`(F, m), i.e., the collection of all m-element subsets of F (combinations without repetition, where order is irrelevant), and retaining only those Φ such that $\phi \in \Phi$. In this way it is similar to the BUFIA algorithm (Chandlee et al., 2019). It is this subroutine that also makes FxF run in time exponential to the data in the worst case. This is because if all features in F are necessary to predict some feature ϕ then `feat_combo` will have to traverse as many subsets as there are in the powerset of $F/\{\phi\}$ before reaching F itself.[1]

However, in the feature system for English, the maximum number of features needed to distinguish each phoneme from every other phoneme is far less than the cardinality of F. BUFIA also has a run time that is exponential in the worst case, but it has proven effective in practice (Swanson et al., 2025; Li, 2025). For these reasons, we do not let the worst-case, exponential time analysis deter our investigation into the practical utility of FxF.

[1]The other aspects of FxF run in polynomial time. Finding the featural projection of the data is linear in the size of the data. Checking the functionality of a projected sample can be done in quadratric time by checking pairs of data points. Literally building the k-ISL DFTs is a polynomial of degree k for a given featural alphabet $|\Sigma_\Phi|$. However, for a finitely many featural alphabets (for instance (V^i) for $1 \leq i \leq n$) these k-ISL DFTs could be stored in advance and just retrieved from memory at the cost of a (large) constant. Finally, SOSFIA itself runs in time linear in the data.

3.3.3 Learning vowel nasalization over features: an example

We illustrate FxF(SOSFIA) with the English vowel nasalization example. Consider the sample S with pairs (mæn, mæ̃n) and (mææ̃, mææ̃). Let F be the feature system in Table 2.3 and $k = 2$. Then FxF(SOSFIA) initializes an empty grammar G, and considers the features in F one by one.

Suppose the first feature it considers is ϕ = [nasal]. It proceeds to run `FeatSearch` with FeatLIST initialized to [{nasal}]. It dequeues FeatLIST, setting $\Phi = \{\text{nasal}\}$ and $m = 1$. It then calculates the projected sample $S_{([\text{nasal}],\text{nasal})}$ to be ([+-+], [+++]) and ([+-+], [+-+]). As such, `is_functional` outputs False, as two same inputs are mapped to different outputs, and `feat_combo` generates all feature sets of size 2, which include [nasal], and places them on NewFeatLIST. `FeatSearch` calls itself again with NewFeatList.

On this next round, the queue FeatLIST contains more than one feature set, and it checks them one at a time. For example, if Φ =[nasal, voice], then `is_functional` would again return False since the projected sample $S_{(\left[\begin{smallmatrix}\text{nasal}\\ \text{voice}\end{smallmatrix}\right],\text{nasal})}$ is not functional. This is because [voice] does not provide sufficient information to predict the right outputs for [nasal]. Particularly, it does not separate vowels from consonants. As such, Φ would be re-assigned to the next element in the queue, and the process would continue.

Eventually, `FeatSearch` dequeues Φ = [nasal, cons]. The projected sample $S_{(\left[\begin{smallmatrix}\text{nasal}\\ \text{cons}\end{smallmatrix}\right],\text{nasal})}$ is functional. Consequently, `FeatSearch` constructs an output empty, k-ISL DFT $\tau_{\square(\left[\begin{smallmatrix}\text{nasal}\\ \text{cons}\end{smallmatrix}\right],\text{nasal})}$ and passes it with $S_{(\left[\begin{smallmatrix}\text{nasal}\\ \text{cons}\end{smallmatrix}\right],\text{nasal})}$ to SOSFIA, returns the output, and terminates.

Interestingly, any feature combination which distinguishes consonants from vowels is sufficient to make the right predictions for the feature [nasal]. For example, the feature [labiodental] assigns a value 0 to all vowels and either $+$ or $-$ to consonants. Consequently, the projected sample $S_{(\left[\begin{smallmatrix}\text{nasal}\\ \text{labdent}\end{smallmatrix}\right],\text{nasal})}$ is functional and would also lead to

SOSFIA outputting a transducer. `FeatSearch` terminates with the first feature combination that works. Thus, the order in which the feature sets Φ are considered does matter, though we abstract away from this choice in this description of the algorithm. We return to this point in the experimental and discussion sections later.

Finally, if none of the feature sets containing [nasal] of length $|\Phi|$ result in `is_functional` outputting True, the length m is increased by one, and the procedure repeats. With the feature [nasal] completed, FxF(SOSFIA) moves on to the next feature, and continues to do so until transducers are obtained for each feature.[2]

To summarize, FxF(SOSFIA) factors the problem of learning a phonological process along featural dimensions. For each feature it searches successively more complex feature combinations until it finds one that generalizes successfully given the other constraints on learning (such as the k-ISL structure). While in the worst case, the problem reaches a stage where it is essentially learning over segmental representations, the idea is that, in practice, the decomposition to features allows correct generalizations to be found more quickly with less data. To establish this point empirically requires a way of measuring the amount of data that is sufficient for FxF(SOSFIA) to succeed and to compare it to the amount of data that is sufficient for SOSFIA over segmental representations to succeed. That is the purpose of the next algorithm.

3.3.4 Characteristic sample and a minimal sample search algorithm (AMßA)

This section describes SOSFIA's characteristic sample (henceforth, CS) that guarantees success in learning k-ISL functions and provides a simple heuristic method for finding a near-minimal characteristic sample that is a subset of the CS provided by Jardine et al. (2014). This heuristic will be used as a measuring stick in the experimental case studies when investigating how much data is sufficient for a successful generalization.

[2] Alternatively, the FxF could be run in parallel for each feature $\phi \in F$.

Jardine et al. (2014) present formal concepts for a class of functions $\mathbb{T}$, a class of representations (grammars) $\mathbb{R}$, a learning algorithm, and a characteristic sample.

Definition 3.6. *Let $\mathbb{T}$ be a class of functions represented by some class of representations $\mathbb{R}$. A sample S for a function $t \in \mathbb{T}$ is a finite set of data for which $(w, v) \in S$ iff $t(w) = v$. A $(\mathbb{T}, \mathbb{R})$-learning algorithm $\mathscr{A}$ is a program that takes a sample for a function $t \in T$ and outputs a representation from $\mathbb{R}$.*

Definition 3.7. *Let $\mathcal{L}$ be a naming, total function that maps $\mathbb{R}$ onto $\mathbb{T}$. For a $(\mathbb{T}, \mathbb{R})$-learning algorithm $\mathscr{A}$, a sample CS is a characteristic sample of a representation $r \in \mathbb{R}$ if for all samples S for $\mathcal{L}(r)$ s.t. $CS \subseteq S$, $\mathscr{A}$ returns r.*

In this context, k-ISL functions are represented with k-ISL DFTs, and SOSFIA outputs a representation r isomorphic to the original k-ISL DFT, which generated the CS. By referring to the properties of k-ISL DFTs and SOSFIA's method of inferring the underlying k-ISL functions, we show that the minimum size of the longest string $s \in \langle \Phi, \Phi' \rangle(CS) = 2k - 1$.

The core idea is that a characteristic sample exposes all possible paths in the transducer to ensure the learning algorithm has sufficient information. In Lemma 16, Jardine et al. (2014) show that for each transition $(q, a, v, r) \in \delta$ in a DFT, it is sufficient if a sample includes a set of strings of the form pas where these symbols are defined as follows.[3]

i. p is the shortest input that leads from the start state to state q

ii. a is the input symbol on the transition

iii. s is a shortest string that leads from state r to any state in the machine

Proposition 1. *For k-ISL functions and for SOSFIA, a sample which includes all strings up to length $2k - 1$ is characteristic.*

[3]There is another condition to Lemma 16, but due to the properties of k-ISL DFTs, it does not apply since the structure of the k-ISL DFTs precludes it.

Proof. From the definition of k-ISL functions, the size of a context that can affect a change is k-long. Therefore, it is necessary to consider k-long substrings from *every* state $q \in Q \setminus \{q_f\}$ in the k-ISL DFT for the function. Denote the shortest string to reach q from q_0 be p_q. We are interested in $\texttt{max}\{p_q \mid q \in Q \setminus \{q_f\}\}$. The furthest non-final state q from q_0 is reachable by a string of length equal to $k-1$. In other words, the depth of the k-ISL machine (excluding q_f) is $k-1$. Hence, the sum of the depth of a k-ISL DFT $(k-1)$ and the memory window (k) is equal to $2k-1$.

By Lemma 16 in Jardine et al. (2014), it follows that if S contains all strings up to length $2k-1$ then it is characteristic. □

Proposition 1 provides a characteristic sample for k-ISL functions, but it does not necessarily provide the smallest such sample. Next we present a procedure for searching for a near-minimal CS: A Minimal Sample Search Algorithm (AMßA). Since AMßA is a greedy, randomized procedure, there is no guarantee that the sample it finds is *the* minimal one. Hence, the output of AMßA is some sufficient subset of the sample in Proposition 1, which will be referred to as *near-minimal CS*.

Algorithm 5: A Minimal Sample Search Algorithm (AMßA)

Data: A segment-based k-ISL DFT τ, a sample $S \subset \rtimes\Sigma^* \ltimes \times\Delta^*$, a feature set F
Result: A near-minimal sufficient sample

```
min_sample ← ∅
min_sample ← min_sample ∪ random(S, 10)
G ← FxF(min_sample, F, k)
while ¬match(G, τ) do
    S ← S \ min_sample
    min_sample ← min_sample ∪ random(S, 10)
    G ← FxF(min_sample, F, k)
end while
return min_sample
```

AMßA is built on top of FxF. It takes three arguments: an original, segmental-based DFT τ representing the target process, a sample S generated with τ, and a feature set F. It selects 10 input-output pairs[4] at random from S and stores them in a variable

[4]We chose 10 as a balance between convenience and fine-grained analysis.

called `min_sample`. The selected pairs are subsequently provided as an argument to FxF(SOSFIA), which ultimately returns a grammar G, which is a collection of k-ISL DFTs, one for each feature in F. AMßA then calls the `match` function (Algorithm 6) to contrast the output of FxF(SOSFIA) with the original τ. For each ϕ in F, the `match` function transforms the original τ by collapsing transitions and states by projecting the output labels under ϕ and by projecting the input labels and state names under the set Φ associated with ϕ in G. This method generates a collection of n distinct DFTs $\tau_{(\Phi,\phi)}$, each corresponding to a feature $\phi \in F$. This set is then evaluated against the output of FxF(SOSFIA) for the same features F.

For example, to evaluate the learned machine $\tau_{([\substack{\text{nasal}\\\text{cons}}],\text{nasal})}$, transitions whose input and output labels correspond to the relevant feature values are extracted from the original machine. Transitions that become identical after projection onto the relevant feature space are then merged. This frequently occurs when distinct segmental transitions correspond to the same feature-level transformation. For instance, the original transitions (ɑ, m, ɑ̃m, m) and (æ, n, æ̃n, n) differ at the fully specified segmental level, but both encode the same feature-level change: an oral vowel is nasalized when followed by a nasal consonant. After projection onto the input features $[\substack{\text{nasal}\\\text{cons}}]$, both transitions therefore collapse into the single transition $([\substack{-}], [\substack{+}], +, [\substack{+}])$.

If the compared transition sets are identical, it means that FxF learned a representation for the target function with `min_sample` and thus AMßA returns it. If not, the procedure continues sampling without replacement until the sample is large enough to infer the target function, i.e. until the sample is *sufficient*.

Algorithm 6: `match`

Data: A learned DFT G, a target DFT τ
Result: Bool

```
δ_t ← transitions in δ ∈ τ
δ_l ← transitions in δ ∈ G
for all (q, σ, v, r) ∈ δ_t do
    if (q, σ, z, r) ∈ δ_l and v ≠ z then
        return False
    end if
end for
return True
```

3.4 Empirical arguments

This section presents four case studies evaluating the performance of FxF(SOSFIA) and SOSFIA in learning four distinct subsequential functions, each representing a phonological process attested in natural language. The processes include: vowel nasalization in English (2-ISL), vowel shortening in Yowlumne (3-ISL), schwa epenthesis in Chukchi (3-ISL), and an interaction between final devoicing and ɔ-raising in Polish (2-ISL and 3-ISL, respectively).

Generally, the objectives of the case studies are to evaluate the performance of the novel algorithm, FxF(SOSFIA). In particular, they test the hypothesis that FxF(SOSFIA) requires significantly less training data compared to SOSFIA. Finally, they also investigate the near-minimal samples produced by AMßA by comparing outputs for both algorithms against the originally generated characteristic samples (CS). Due to the inherent randomness in AMßA's minimal sample selection, each experiment was repeated ten times. The reported results include both the mean and standard deviation across the runs.

The characteristic samples were constructed using a subset of phonemes from the respective languages. Ideally, the entire phonemic inventories would be used. However, full data generation proved computationally expensive due to the size of the

resulting CSs. For example, Yowlumne's inventory includes 43 segments, and modeling vowel shortening as a 3-ISL function yields a characteristic sample of size $|CS| = 150{,}508{,}643$ (see Proposition 1). To keep the experiments tractable, phonemic inventories were reduced such that $|CS|$ did not exceed 20,000. However, this practical decision was not without consequences, as discussed in later sections.

All of the data and code used to run these experiments are available at the author's github repository.

3.4.1 Representative k-ISL functions

This section presents five phonological processes attested across four natural languages. The processes were deliberately selected to illustrate variation in the value of k as well as in the number and type of phonological features involved in their computation. As with the case of English vowel nasalization, we acknowledge that alternative analyses and debates exist for some of the phenomena discussed. Our aim is not to argue for any one analysis in particular but instead to use concrete examples to meet our objectives. These examples should be understood as representative instances of broader classes of processes that fall within the expressive range of k-ISL functions.

Vowel nasalization in English

Vowel nasalization was introduced and discussed in section 2.4.3. As mentioned, the process has been expressed with a phonological rule like the one shown in (2.4.3), repeated below.

(2) Vowel Nasalization (VN)

$$[-\text{cons}] \longrightarrow [+\text{nasal}] \;/\; \_\_ \left[\begin{smallmatrix} +\text{cons} \\ +\text{nasal} \end{smallmatrix} \right]$$

By modeling vowel nasalization in this way, we test whether a learner can infer a contextually conditioned alternation that depends on the feature composition of a following segment. In particular, we expect that the feature [nasal] will not be sufficient

on its own to learn the generalization and will require additional information from other features, such as [consonantal], which condition the alternation. The phonemes used for the experiments are presented in Table 2.3.

Vowel Shortening in Yowlumne

Yowlumne (also called Yawelmani) is a dialect of Yokuts, an Indigenous North American language spoken in California. Its phonemic inventory consists of consonants exhibiting contrastive aspiration and glottalization, and vowels with underlying length contrast (Hockett, 1967; Kenstowicz, 1994). The 7 phonemes used to generate the dataset are summarized in Table 3.2.

Segment	o	i	o:	i:	t^h	g	n
high	-	+	-	+	0	+	0
low	-	-	-	-	0	-	0
tense	-	+	-	+	0	0	0
front	-	+	-	+	0	0	0
back	+	-	+	-	0	0	0
round	+	-	+	-	-	-	-
long	-	-	+	+	0	0	0
cons	-	-	-	-	+	+	+
son	+	+	+	+	-	-	+
cont	+	+	+	+	-	-	-
delrel	0	0	0	0	-	-	0
approx	0	0	0	0	-	-	-
nasal	-	-	-	-	-	-	+
voice	+	+	+	+	-	+	+
lab	0	0	0	0	-	-	-
labdent	0	0	0	0	-	-	-
cor	0	0	0	0	+	-	+
ant	0	0	0	0	-	0	+
distr	0	0	0	0	-	0	-
strid	0	0	0	0	-	0	-
lat	0	0	0	0	-	-	-
dor	0	0	0	0	-	+	-
constrgl	0	0	0	0	+	-	-
spreadgl	0	0	0	0	-	-	-

TABLE 3.2: Feature chart for Yowlumne phonemes used in experiments.

Yowlumne exhibits a process of vowel lengthening, where underlying long vowels are shortened on the surface. This alternation can be observed, for instance, in imperative and non-future verb stems (Kenstowicz, 1994, p. 108).

(3)

future	imperative	non-future	gloss
wo:n-en	won-ko	won-hin	'hide'
la:n-en	lan-ka	lan-hin	'hear'
me:k-en	mek-ka	mek-hin	'swallow'

Kenstowicz and Kisseberth (1979)[p. 84] propose that the condition for the vowel shortening is two consecutive consonants.

(4) Vowel Shortening (VS)

$\left[\begin{smallmatrix}-\text{cons}\\+\text{long}\end{smallmatrix}\right] \longrightarrow$ [-long] / ___ [+cons][+cons]

We adopt Kenstowicz and Kisseberth's rule in (4), since it provides a clear view into the contrast between 2-ISL and 3-ISL function learning where the change affects only a single feature.[5] We expect that [long] alone does not carry sufficient information to infer the underlying function on its own.

Schwa epenthesis in Chukchi

Chukchi is an endangered Chukotko-Kamchatkan language spoken in Siberia. From its phonemic inventory of five vowels and 13 consonants (Dunn, 1999), seven were chosen for experiments (Table 3.3).

Chukchi was selected for this study because it exemplifies a different type of phonological pattern, namely, final ə-epenthesis (see examples in 5). This case is particularly interesting because, unlike in the previous processes, this transformation affects every single feature and predicting sufficient featural combinations for learning is not an immediate and trivial task. In English, for example, it was evident that [nasal] depended on contextual information from [cons]; here, on the other hand, the specifications required for successful learning of individual features are far less transparent.

[5] We acknowledge that there are alternative analyses of vowel shortening in Yowlumne. For instance, Archangeli (1991) analyzes these alternations as an emergent consequence of prosodic well-formedness and Blevins (2004) argues that many length alternations are morphological or lexical in origin.

Segment	i	o	m	p	t	ɬ	s
high	+	-	0	0	0	0	0
low	-	-	0	0	0	0	0
tense	+	+	0	0	0	0	0
front	+	-	0	0	0	0	0
back	-	+	0	0	0	0	0
round	-	+	-	-	-	-	-
long	-	-	0	0	0	0	0
cons	-	-	+	+	+	+	+
son	+	+	+	-	-	-	-
cont	+	+	-	-	-	+	+
delrel	0	0	0	-	-	+	+
approx	0	0	-	-	-	-	-
nasal	-	-	+	-	-	-	-
voice	+	+	+	-	-	-	-
lab	0	0	+	+	-	-	-
labdent	0	0	-	-	-	-	-
cor	0	0	-	-	+	+	+
ant	0	0	0	0	+	+	+
distr	0	0	0	0	-	-	-
strid	0	0	0	0	-	-	+
lat	0	0	-	-	-	+	-
dor	0	0	-	-	-	-	-

TABLE 3.3: Feature chart for Chukchi phonemes used in experiments.

Krause (1980) identifies three distinct environments in which /ə/ may be inserted, typically to break up biconsonantal or triconsonantal clusters. For the purposes of this paper, we only focus on the insertion of schwa to break up word-final bi-consonantal clusters. Examples of this phenomenon are shown below.

(5)

noun-ABS.SG	**noun-ABS.PL**	**gloss**
mimәl	mimlәt	'water'
lewәt	lewtәt	'mouth'
qepәl	qeplәt	'ball'

In these example, the underlying representations arguably are /miml, lewt, qepl/ for the singular forms and /mimlt, lewtt, qeplt/ for the plural forms. A rule like the one in (6) applies to break up the word-final consonant cluster.

(6) Final epenthesis (FE)

ø⟶ ə / [+cons]__[+cons] #

We acknowledge that the distribution of epenthetic schwa in Chukchi is more complex than modeled in (6), as it may interact with additional phonological processes,

including final vowel deletion (Dunn, 1999). Our goal, however, is to test the effectiveness of the FxF(SOSFIA) on this type of process. That is the reason why we restrict the process to one environment: the insertion of /ə/ between two consonants at a word boundary (Krause, 1980, p. 53).

Final devoicing and o-raising in Polish

The final phonological phenomena that we consider involves interaction of two strictly local processes. Rule interaction has a rich history in phonological theory (Kiparsky, 1973; Baković, 2007; Chandlee et al., 2018; Baković and Blumenfeld, 2024).

In this study, we focus on an *opaque* interaction between two processes in Polish: word final obstruent devoicing and ɔ-raising. Data in 7 presents selected cases of /ɔ/ raising to [u] when followed by voiced obstruents and approximants and remaining the same in the context of a following nasal.

(7)

singular	**plural**	**gloss**
ruk	rɔg-i	'horn'
sɔk	sɔk-i	'juice'
ʐ̨wup	ʐ̨wɔb-i̇	'crib'
snɔp	snɔp-i̇	'sheaf'
bur	bɔr-i̇	'forest'
dzwɔn	dzwɔn-i̇	'bell'
dɔm	dɔm-i̇	'house'

The interaction is called opaque because there are forms like [ruk] 'horn, sg.' and [ʐ̨wup] 'crib, sg.' where ɔ-raising has applied yet the following sound in the surface form is devoiced. Typical analyses, discussed below, account for this by applying ɔ-raising before word-final obstruent devoicing.

Similarly to the previous cases, we acknowledge that the ɔ-raising process is complex and has been the subject of various analyses (see Bethin 1978; Buckley 2001; Gussmann 2007). Here we adapt the SPE style analysis of the rule described in Kenstowicz

(1994, p. 77). To correctly derive the surface forms, ɔ-raising (8) must precede final obstruent devoicing (9). This creates a counterbleeding type of order as the reverse order would prevent ɔ-raising from applying.

(8) Final Devoicing (FD)

[-sonorant] $\longrightarrow$ [-voice] / __ $\ltimes$

(9) ɔ-raising (OR)

$\begin{bmatrix} -\text{cons} \\ +\text{back} \\ -\text{low} \end{bmatrix} \longrightarrow$ [+high] / __ $\begin{bmatrix} +\text{cons} \\ +\text{voice} \\ -\text{nasal} \end{bmatrix} \ltimes$

For example, consider the underlying representation of 'horn' /rɔg/. OR applies first yielding an intermediate form [rug]. Then FD applies yielding the surface form [ruk]. On the other hand, if FD applied first it would yield *[rɔk], to which OR would not apply because the final obstruent of this intermediate form is voiceless, predicting an incorrect surface form.

FD is a 2-ISL function while OR is 3-ISL. Their ordered composition (OR then FD) can be modeled with a 3-ISL function, call it CB (for counterbleeding). The finite-state representation of CB does not literally execute OR and then FD; for instance there is no intermediate representation [rug] that appears in any execution of CB (see Beesley and Karttunen (2003) for additional discussion regarding this point). Interestingly, as it will become apparent in the following section learning over features naturally separates these processes. Chandlee et al. (2018) provide additional discussion regarding how $k-$ISL DFTs can model a range of processes which interact opaquely.

The phonemes chosen for the experiments are presented in 3.4.

3.4.2 Data generation

For each of the four k-ISL functions, a corresponding k-ISL DFT is constructed and a characteristic sample consisting of underlying-surface form pairs is generated accordingly. Following the Proposition 1, a set of all logically possible n-grams over Σ^*

Feature	a	ɔ	ɔ̃	u	b	t	d	n	ʂ	ʐ
voice	+	+	+	+	+	-	+	+	-	+
son	+	+	+	+	-	-	-	+	-	-
cons	-	-	-	-	+	+	+	+	+	+
cont	+	+	+	+	-	-	-	-	+	+
cor	0	0	0	0	-	+	+	+	+	+
ant	0	0	0	0	0	+	+	+	-	-
dor	0	0	0	0	-	-	-	-	-	-
lab	0	0	0	0	+	-	-	-	-	-
distr	0	0	0	0	0	-	-	-	-	-
delrel	0	0	0	0	-	-	-	0	+	+
nasal	-	-	+	-	-	-	-	+	-	-
lat	0	0	0	0	-	-	-	-	-	-
high	-	-	-	+	0	0	0	0	0	0
back	+	+	+	+	0	0	0	0	0	0
low	+	-	-	-	0	0	0	0	0	0
tense	-	-	-	+	0	0	0	0	0	0
round	-	+	+	+	0	0	0	0	0	0

TABLE 3.4: Feature chart for Polish phonemes used in experiments.

up to length $2k - 1$ is created to serve as the set of potential underlying forms.[6] The constructed k-ISL DFTs are then used to generate the corresponding surface forms (outputs). These same transducers are also used to evaluate the results.

As previously mentioned, a constraint was imposed on the total size of each characteristic sample (CS) to ensure that the number of generated input–output pairs did not exceed 20,000. Since the size of a CS, $|CS|$, depends on both the size of the alphabet ($|\Sigma|$) and the function's k value, these parameters were adjusted accordingly. For the 2-ISL function, $|\Sigma| = 26$, resulting in a CS of 18,287 pairs. For the 3-ISL functions, a reduced alphabet size of $|\Sigma| = 7$ was used, yielding CSs of 17,206 input-output pairs. We ran additional experiments on Polish, where the size of the input alphabet was increased up to 10 to further showcase the difference between the growth of CS when learning over segments vs. features (see Table 3.14 for details).

We introduce terminology corresponding to each of the types of samples generated. There are five sample types performing various roles, which are listed below.

[6]The generated data is constructed to be extensionally consistent with the four functions, but it is not intended to represent attested or naturalistic linguistic patterns. Learning with naturalistic data will be addressed in the next stage of this research.

- a sample (**CS**): a characteristic sample over segmental representations
- a near-minimal *segmental* sample (**M**): a near-minimal sample randomly drawn from CS with AMßA calling SOSFIA.
- a feature-based sample ($\mathbf{S}_{(\Phi,\phi)}$): a sample extracted from CS whose inputs and outputs are projected according to Φ and ϕ, respectively, for each feature $\phi \in F$. For example, $\mathrm{S}_{(\text{[high],high})}$ is a sample extracted from CS to predict [high] from [high], while $\mathrm{S}_{(\left[\begin{smallmatrix}\text{high}\\ \text{round}\end{smallmatrix}\right],\text{high})}$ means that two features [high] and [round] are projected to predict [high].
- a near-minimal feature sample ($\mathbf{M}_{(\Phi,\phi)}$): a near-minimal feature sample randomly drawn from $S_{(\Phi,\phi)}$ with AMßA calling the original SOSFIA algorithm.
- a synthesis of $M_{(\Phi,\phi)}$ for all $\phi \in F$ ($\mathbf{M_F}$): a segmental sample reconstructed by combining the minimal feature-based samples $M_{(\Phi,\phi)}$ for each feature $\phi \in F$. To estimate how many full segmental input-output pairs can be formed from these minimized components, we identify compatible combinations of feature vectors that jointly satisfy subsets of the segmental transformations observed in CS. For example, the following feature-based data points

$$
\begin{aligned}
&\ldots\\
&\left(\left[\begin{smallmatrix}-+-\\ ++-\end{smallmatrix}\right], [++-]\right) \in S_{(\text{[nasal, cons],cons})},\\
&([0+-],[0+-]) \in S_{(\text{cor,cor})},\\
&([0--],[0--]) \in S_{(\text{lat,lat})},\\
&\ldots
\end{aligned}
$$

are all subcomponents of a segmental pair $(\mathrm{ant}, \tilde{\mathrm{a}}\mathrm{nt}) \in$ CS. The synthesized set $\mathbf{M_F}$ reflects an *upper bound* on the number of segmental input–output pairs that can be reconstructed from the featural data.

3.4.3 Results

This section presents a summary and interpretation of the results. First, as expected, all four functions were successfully learned from the characteristic sample, both with SOSFIA and with FxF(SOSFIA). Second, when the data sample input to the learning algorithms was reduced with AMßA, FxF(SOSFIA) consistently outperformed the segmental baseline SOSFIA in terms of the amount of data sufficient for correct generalization. The advantage of FxF was especially pronounced in cases involving a single feature change and in the epenthesis process. The Polish case, which involved interaction of multiple features, required additional justification and further experimentation to confirm the effectiveness of the approach. Third, the results demonstrate that the data reduction obtained by AMßA for FxF(SOSFIA) is more substantial than what AMßA obtains for SOSFIA. For the feature-based FxF(SOSFIA) the reductions in size from the CS ranged from 98% in the best case to 52% in the worst. In contrast, for the segment-based SOSFIA, AMßA only achieved a reduction ranging from 66% to just 0.15%. We report here only the results most relevant to the main discussion; full experimental details and quantitative data are provided in Appendices A, B and C.

In the ensuing discussion of the learning results, we refer to features in F as either *sufficient* or *insufficient*. A feature ϕ is termed insufficient if its values alone do not provide enough information to infer the target function. Formally, this means the sample $S_{(\{\phi\},\phi)}$ is not functional and thus `FeatSearch` must add additional features to yield a functional sample. A feature ϕ is termed sufficient if it is not insufficient (and so the sample $S_{(\{\phi\},\phi)}$ is functional).

As mentioned in section 3.3.3, it is often true that multiple feature combinations

are able to successfully predict a feature ϕ. While `FeatSearch` is defined to stop once the first one is found, our implementation returned all feature sets with the smallest cardinality capable of predicting ϕ since we were curious about them. As such, when calculating the synthesized sample M_F, we used the combination which yielded the smallest functional sample for each insufficient feature (or randomly selected one if there were more than one equally smallest samples). While this technique was adequate to demonstrate the core concept proposed here, future research will explore more linguistically informed methods for selecting appropriate featural combinations in cases where individual features are insufficient for learning (see Section 3.5 for more details).

English

Of the 22 features used in the English vowel nasalization case study, only one feature, particularly [nasal], was found to be insufficient, as expected. The remaining 21 features were individually sufficient and yielded functional samples on their own. The projected samples $S_{(\Phi,\phi)}$ for these features varied in size depending on the number of values ϕ takes. For instance, as shown in Table 2.3, the binary feature $\phi = [\text{voice}]$ (with values $\{+, -\}$) yielded a sample $S_{(\text{voice,voice})}$ comprising 14 input-output pairs. In contrast, the ternary feature [cor] produced $S_{(\text{cor,cor})}$ with 39 input-output pairs. These samples were successfully reduced using AMßA to average sizes of $M_{(\text{voice,voice})} = 9$ and $M_{(\text{cor,cor})} = 23$, respectively. Table B.1 provides the average values of $M_{(\Phi,\phi)}$ across all features, and Table B.7 provides complete results corresponding to each individual feature.

We identified 10 feature combinations yielding functional samples for the insufficient feature [nasal], whose sizes also varied depending on the size of the resulting featural alphabets Σ_Φ. As was already presented in Figure 3.1, combination of features [nasal, cons] resulted in the input alphabet $\Sigma_{\left[\begin{smallmatrix}\text{nasal}\\ \text{cons}\end{smallmatrix}\right]} = \{[\begin{smallmatrix}-\\-\end{smallmatrix}], [\begin{smallmatrix}-\\+\end{smallmatrix}], [\begin{smallmatrix}+\\+\end{smallmatrix}]\}$. Recall that the

remaining logical feature value $[\pm]$ was not considered as we assume English does not have underlying nasal vowels.

Accordingly, the projected sample $S_{([\substack{\text{nasal}\\\text{cons}}]),\text{nasal})}$ contained 39 pairs. Other combinations involving [nasal] produced larger sample sizes, particularly 84 and 155 pairs, depending on whether the featural alphabet contained four or five elements, respectively. In all cases, AMßA was able to reduce the samples substantially. Since the pair [nasal, cons] yielded the smallest functional sample, we used this combination to construct the synthesized featural sample M_F.

Feature set Φ	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	**Avg.** $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	**SD**
[**voice**]	14	9	2.66
[**cor**]	39	23	4.76
[**nasal**, cons]	39	31	3.89
[**nasal**, front]	84	68	10.9
[**nasal**, approx]	155	117	16.22

TABLE 3.5: Selected representative samples $|S_{(\Phi,\,\phi)}|$, Avg. $|M_{(\Phi,\,\phi)}|$ and standard deviation (SD) for VN in English for features [voice], [coronal], and [nasal].

Table 3.5 presents a sample of cardinalities of the extracted samples for features ϕ ([voice], [coronal], and [nasal], particularly). The Feature Sets Φ, in the first column, here as well as in the corresponding tables for the remaining test cases, indicate what features were used for input projections for ϕ. The bolded feature represents the specific feature ϕ whose values are predicted. When only a single bolded feature appears, such as [**voice**], it means that the values of that feature alone were sufficient to define the characteristic sample for the corresponding feature. In cases where multiple features appear, as in [**nasal**, cons], the bolded feature (ϕ) is the (insufficient) predicted feature, while the full set in brackets represents Φ, the set of features that give a functional sample for ϕ.

Given the data samples $M_{(\Phi,\phi)}$ selected by AMßA for each $\phi \in F$, we reconstructed the corresponding segmental sample M_F. This enabled direct comparison with the

segmental characteristic sample CS and its averaged AMßA-reduced counterpart M.

|CS|	Segments		Features	
	Avg. |M|	SD	Avg. $\|M_F\|$	SD
18,278	6,157	650.21	188	10.06

TABLE 3.6: Averaged segmental sample $|M|$ and averaged synthesized featural $|M|_{\mathrm{F}}$ and their corresponding SDs in English.

The average size of a reduced characteristic sample M derived from the segmental representation was 6,157. In contrast, the FxF learner identified significantly smaller successful samples, with M_F averaging only 188 elements. Starting from an original sample of $|CS| = 18{,}278$, segmental learning achieved a 66% reduction, whereas featural learning achieved a 98% reduction. This substantial difference demonstrates that featural decomposition in the FxF framework yields more compact and data-efficient generalizations.

Yowlumne

This case study examines vowel shortening in Yowlumne, a process that, like English vowel nasalization, targets a single feature: [long]. What differs is that the environment for the change is larger and thus requires more memory.

Contrary to our expectations, *all* features were sufficient for learning the vowel shortening function, including [long] in isolation. This finding reflects a representational asymmetry: the values assumed for the feature [long] themselves encode distributional distinctions that parallel the relevant contextual split. In our dataset, [long] takes on ternary values: {+, –, 0}, where {+, –} are exclusively associated with vowels, and 0 with consonants. As a result, the learner can indirectly infer the necessary contextual distinctions purely from this feature's values. This outcome show that not only the featural decomposition but also the range and interpretation of values, can play a decisive role in learnability.

Table 3.7 presents results from AMßA's reductions of individual feature-based samples $S_{(\Phi,\phi)}$. In the best-performing case, the sample for [cor] was reduced by 59%. The smallest reduction was observed for the target feature [long], which is expected, as it is the only feature that changes and, therefore, must be represented with particular data that cover the necessary paths in the transducer.

Feature set Φ	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	**Avg.** $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	**SD**
[**voice**]	54	33	10.11
[**round**]	62	30	7.39
[**ant**]	309	136	19.28
[**cor**]	336	139	24.05
[**long**]	363	317	49.02

TABLE 3.7: Selected representative samples $\lvert S_{(\Phi,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\phi)}\rvert$ and standard deviation (SD) for VS in Yowlumne for features [voice], [round], [anterior], [coronal], and [long].

Interestingly, the projected sample sizes $S_{(\Phi,\phi)}$ varied even among features with identical value sets. For example, both [voice] and [round] are binary features with values {+, –}, yet their sample sizes differed. This disparity is attributable to the distribution of feature values among the segments. Because $S_{(\Phi,\phi)}$ is projected from CS, the features whose values were less evenly distributed among the segments resulted in smaller projected samples. For example, in our data, only one of the seven segments was marked as [–voice], whereas two segments were [–round]; this skew resulted in a smaller sample for [voice].

$\lvert\mathbf{CS}\rvert$	**Segments**		**Features**	
	Avg. $\lvert\mathbf{M}\rvert$	**SD**	Avg. $\lvert\mathbf{M}_F\rvert$	**SD**
17,206	16,583	702.06	952	49.90

TABLE 3.8: Averaged segmental sample $\lvert M\rvert$ and averaged synthesized featural $\lvert M\rvert_{\text{F}}$ and their corresponding SDs in Yowlumne.

Overall sample sizes are summarized in Table 3.8. The segmental sample M, derived without feature factorization, had an average size of 16,583 (4% reduction). In

contrast, the synthesized featural sample M_F required only 952 examples on average. The larger transducer sizes observed in the Yowlumne case, driven by the larger k value, contributed to the higher sample size, which was expected as discussed in Section 3.2. Nevertheless, the difference between reduction over segmental learning vs. featural is much more pronounced than in the English case study.

Chukchi

The Chukchi epenthesis function presents a different learning scenario, where each feature is affected due to the inserted segment. Surprisingly, of the 22 features used to represent Chukchi segments, 13 were *sufficient* on their own. These tended to be features that, similar to the situation in Yowlumne, implicitly distinguished vowels from consonants via their value patterns. For instance, [labial] assigns the value 0 to vowels and either $+$ or $-$ to consonants, providing a natural partition of the inventory. The remaining 9 features were *insufficient* individually, but when paired with complementary features that supplied the missing contrast, they successfully learned the target function. For example, while [sonorant] alone does not separate vowels from consonants, its combination with [nasal] provides enough information to infer the correct generalization.

Feature set Φ	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	**Avg.** $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	**SD**
[**tense**]	62	32	5.69
[**high**]	309	165	30.34
[**round**, high]	309	225	30.28
[**lab**]	363	313	8.88
[**cont**, tense]	363	306	18.02
[**strid**, dor]	1,236	955	58.46

TABLE 3.9: Selected representative samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for FE in Chukchi for features [tense], [high], [round], [labial], [continuant], and [strident].

Sample sizes for the projected datasets $S_{(\Phi,\phi)}$ varied widely, and in many cases were substantially larger than those in the English and Yowlumne experiments. For

example, the combination $S_{(\left[\begin{smallmatrix}\text{strid}\\ \text{dor}\end{smallmatrix}\right],\text{strid})}$ produced 1,236 input-output pairs due to the resulted 4-element alphabet $\Sigma_{\left[\begin{smallmatrix}\text{strid}\\ \text{dor}\end{smallmatrix}\right]}$. Table 3.9 shows the reductions obtained with AMßA.

Despite the challenge posed by the global nature of the epenthesis transformation, FxF learning remained robust.

|CS|	**Segments**		**Features**	
	Avg. |M|	**SD**	Avg. $\|M_F\|$	**SD**
17,206	17,181	28.87	2,331	46.27

TABLE 3.10: Averaged segmental sample $|M|$ and averaged synthesized featural $|M|_{\text{F}}$ and their corresponding SDs in Chukchi.

Table 3.10 summarizes the overall segmental and featural sample sizes. The reduced characteristic segmental sample M remained nearly identical in size to the original CS (avg. $|M| = 17{,}181$), showing that little to no compression was achieved without factorization. In contrast, the FxF approach yielded a significantly smaller sample of 2,331 input-output pairs on average. Although this is larger than the feature-based sample size obtained for the Yowlumne function (avg. $M_F = 952$), the difference reflects the more complex nature of the process (feature insertion vs. feature changing). In this context, the observed 86% reduction remains a strong result, demonstrating the scalability and effectiveness of featural decomposition.

Polish

The Polish case study presents 3-ISL function involving two phonological processes in a counterfeeding relation. While the featural learner remained successful, the degree of sample reduction was significantly less than what was found with the English, Yowlumne, and Chukchi case studies. This was not due to the function representing two processes simultaneously, but rather due to the complexity of one of them; specifically, the transformation of /ɔ/ to [u], which required coordinated information from multiple features.

In this alternation, features such as [high] and [tense] were insufficient on their own and required combination with three additional features to produce functional samples. Unlike the processes that target broader classes (e.g. English [+nasal, –cons]) or Yowlumne (e.g., [+long, –cons]), the ɔ-raising processes in Polish targets a single segment, which necessitates distinctions between similar segments, here vowels. For instance, [low] is needed alongside [high] to exclude vowel /a/, which does not undergo change. Contextual triggers, on the other hand, could be successfully encoded by features like [voice] and [sonorant].

Feature set Φ	$\lvert S_{(\Phi,\phi)}\rvert$	**Avg.** $\lvert M_{(\Phi,\phi)}\rvert$	**SD**
[**nasal**]	54	29	14,92
[**son**]	62	27	12.77
[**round**]	336	137	51.76
[**voice** son]	336	214	10.73
[**high** voice son low]	8,250	7,119	907.70
[**tense** voice son round]	8,250	7,372	806.83

TABLE 3.11: Selected representative samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for FD and OR in Polish for $\lvert\Sigma_{\text{seg}}\rvert = 7$, for features [nasal], [sonorant], [round], [voice], [high], [tense].

Table 3.11 above presents a subset of representative feature combinations for $|\Sigma| = 7$. While most projected featural samples $S_{(\Phi,\phi)}$ were reduced by AMßA, the reduction was less prominent for features that required extensive combinations. In particular, the input alphabet for feature [high] ($V_{[\text{high,voice,son,low}]}$) was reduced only marginally, reflecting the difficulty of compressing transformations involving singleton segments. As a result, the projected sample contained 8,250 input-output pairs and was reduced only by approximately 14%.

Because the reduced samples $M_{(\Phi,\phi)}$ were significantly large, the averaged M_F samples remained large as well. Nevertheless, when compared to reduction over segmental data representation, FxF still comes out ahead. Table 3.12 shows that the FxF

|CS|	Segments		Features	
	Avg. |M|	**SD**	Avg. $\|M_F\|$	**SD**
17,206	15,169	2,063.50	8,235	208.39

TABLE 3.12: Averaged segmental sample $|M|$ and averaged synthesized featural $|M|_{\mathrm{F}}$ and their corresponding SDs in Polish.

procedure achieved 52% reduction with AMßA, as opposed to the reduction AMßA obtained with segmental representations, which was 12%.

Feature set Φ	$\|\Sigma_{\text{seg}}\| = 8$		$\|\Sigma_{\text{seg}}\| = 9$		$\|\Sigma_{\text{seg}}\| = 10$	
	$\|S_{(\Phi,\phi)}\|$	$\|M_{(\Phi,\phi)}\|$	$\|S_{(\Phi,\phi)}\|$	$\|M_{(\Phi,\phi)}\|$	$\|S_{(\Phi,\phi)}\|$	$\|M_{(\Phi,\phi)}\|$
[**nasal**]	54	43	62	29	62	43
[**son**]	62	32	62	34	62	23
[**round**]	336	162	336	136	336	153
[**voice** son]	363	241	363	223	363	235
[**high** voice son low]	8,446	7,416	-	-	-	-
[**tense** voice son round]	8,446	6,656	-	-	-	-
[**high** voice nasal low]	-	-	17,892	13,122	17,892	8,122
[**tense** voice nasal round]	-	-	17,892	12,482	17,892	17,842

TABLE 3.13: Selected representative samples $|S_{(\Phi,\,\phi)}|$ and $|M_{(\Phi,\,\phi)}|$ FD and OR in Polish with increasing $|\Sigma_{\text{seg}}|$ of 8, 9 and 10 phonemes.

To evaluate the robustness of the feature-based learner under increasing inventory size, we incrementally expanded the phoneme set by one symbol at a time ($|\Sigma_{\text{seg}}| = 8, 9, 10$) and re-ran the experiments for each resulting alphabet size. Table 3.13 shows that although the projected $S_{(\Phi,\phi)}$ samples grew in size, AMßA continued to yield considerable reductions. Table 3.14 reports the overall segmental and featural reductions at the full-sample level. While the segmental learner benefited modestly from data pruning (at best 21% reduction with $|\Sigma_{\text{seg}}| = 10$), the feature-based learner achieved far more efficient compression. Particularly, with the largest alphabet, the reduction peaked to 82%. These results suggest that even in cases involving tightly constrained

alternations and rare, singleton targets, FxF approach remains scalable and resilient to inventory growth.

$\lvert\Sigma_{\text{seg}}\rvert$	$\lvert\text{CS}\rvert$	**Segments** $\lvert\text{M}\rvert$	**Features** $\lvert\text{M}_F\rvert$
8	33,352	30,344	8,283
9	59,868	51,161	16,272
10	101,110	79,540	17,945

TABLE 3.14: Samples |S|, |M|, and $|\text{M}|_{\text{F}}$ for increasing $|\Sigma|_{\text{seg}}$ of 8,9, and 10 phonemes in Polish.

This is particularly noticeable for feature [high] in the $|\Sigma_{\text{seg}}| = 10$ condition. While the segmental alphabet is of size 10, the features used to predict [high] yield an inventory of 7. We suspect that given the full inventory of Polish phonemes, the projected sample sizes for [high] will remain approximately same as to when $|\Sigma_{\text{seg}}| = 10$. More generally, although increasing $|\Sigma_{\text{seg}}|$ will continue to enlarge the segmental sample, the featural alphabet $|\Sigma_{\Phi}|$ sizes eventually saturate: once the inventory already realizes all relevant combinations of feature values, adding additional segments does not introduce any new featural patterns, and thus does not increase $|\Sigma_{\Phi}|$.

3.5 Discussion

The empirical studies presented in Section 3.4 demonstrate that the Factor-by-Feature (FxF) approach significantly enhances the learnability of k-ISL functions by reducing the size of the required training samples. By introducing a systematic featural decomposition, FxF enables learners to utilize the inherent structure of phonological systems, which in turn reduces the complexity and size of the training sample requirement. The success of this approach can be consistently observed across a range of phonological processes, where feature-based learner outperforms the baseline segmental learner in data efficiency.

A key contribution of FxF is the modularization that it brings. This approach allows to factor the learning problem into smaller, feature-specific subproblems, which leads to efficient reduction of the underlying transducers. For processes involving a single feature alternation (e.g., English vowel nasalization, Yowlumne vowel shortening), the improvement is especially pronounced, with sample size reductions of up to 98%. Even in more complex settings like Polish, where rule interaction and singleton transformations make learning more challenging, FxF still achieved a substantial reduction (over 50%) relative to the original sample size.

Another important contribution is the stability of featural sample sizes under increased alphabet size. As shown in the Polish experiments, the size of featural samples grew much more slowly than segmental ones, highlighting the scalability of the approach. This is crucial for practical applications, especially when working with large phonemic inventories or naturalistic data.

It is also worth noting that the alternating features in Polish naturally separate the two processes: in ɔ-raising, the change is carried by features [tense] and [high] (with [voice] and [sonorant] appearing only in the triggering context), whereas in word-final devoicing [voice] itself is the feature that changes. This indicates that there are circumstances where FxF can track distinct processes that invites future investigation.

Taken together, these findings support the hypothesis that phonological learning benefits not only from prior knowledge about the type of function (e.g. k-ISL), but also from the representational flexibility allowed by featural decomposition. The FxF method leverages this flexibility to construct more compact and learnable representations. This offers a promising direction for computational models of phonological acquisition and phonological processes in low-resource languages, while remaining interpretable and grounded in grammatical-inference methods with theoretical guarantees.

Despite the advantages of incorporating featural representation in learning classes of subsequential functions, the FxF approach could be further advanced in various

ways. First, while the present study holds the memory window k constant and assumes the structure of the underlying transducer, the empirical results suggest that k, in many cases, can be minimized for individual features. Particularly, the features that do not reflect any changes between the inputs and outputs could be, in theory, learned assuming a much simpler underlying structure, i.e. 1-ISL DFT. Second, the current work consistently assumes left-to-right processing of the input string. Although it is known that the direction of processing does not affect the structure of a k-ISL DFT, the potential effects of directionality on learnability have not yet been explored. For most of the functions discussed here, the necessary context for change was on the right, which requires the use of 'waiting transitions', i.e. transitions that output λ (the empty string), postponing output until the relevant context falls within the k window. If, instead, processing direction was reversed to *right-to-left*, the relevant context would be seen before the target symbol, eliminating the need for such waiting transitions. Such change in directionality can thus reduce the number of transitions whose outputs differ from the corresponding inputs. Under the default assumption of input-output identity at the beginning of learning process, switching the direction of processing may lead to further reduction of characteristic samples. Both of those shortcomings are addressed in Chapter 4.

Third, data in this study were synthetically generated. To more rigorously evaluate the FxF approach, future work should incorporate naturalistic data. An immediate first step would be to apply the model to problem sets from phonology textbooks, which provide more realistic, yet curated, examples which respect different components of phonological grammar, such as phonotactic constraints. This challenge is addressed in Chapter 6 together with the newly proposed string extension learner, SEAL, introduced in Chapter 5. Ultimately, extending the analysis to include data found in the corpora of child-directed speech will allow for a more thorough assessment of the approach's plausibility as a model of human language acquisition or language variation. We leave that for future work.

Fourth, the current study is limited to k-ISL functions, which while capturing a broad class of phonological processes, do not cover *all* types of attested phenomena. Extending the framework to capture more complex processes, such as long-distance or tonal, will require adapting algorithms that handle tier-based representations, such as k-OSL functions (Burness and McMullin, 2019; Burness et al., 2021), or the tier definite and reverse definite classes (Lambert and Heinz, 2024). Investigating these extensions is necessary for building models that are both typologically comprehensive and empirically grounded. Consequently, Chapter 7 extends the framework to tier-based representations to model long-distance processes.

Chapter 4

K-value and directionality

In the previous chapter, we introduced the main approach taken in this dissertation for learning from small datasets, focusing on the use of phonological representations. We showed that features can substantially reduce the amount of data required for successful learning. In the framework and experimental conditions adopted here, comparable reductions are not observed for the segment-based baseline learner, mainly because the larger segmental alphabet induces a much larger transducer hypothesis space.

In this chapter, we extend that approach with two additional techniques aimed at further reducing the size of the characteristic sample needed for learning. The first is a simple method for learning the k value in k-ISL functions separately for each feature. Many features in a process remain unchanged and therefore can be modeled with a smaller k (often $k = 1$), which allows them to be learned from only a few examples or even none, given the default assumption of input–output identity. The second technique involves changing the direction in which the input string is processed. While the direction does not alter the formal structure of a k-ISL function, it can have important consequences for learning: processing from right to left for processes triggered by right-hand context can eliminate the need for λ transitions, the transitions where the output is temporarily halted until more context is revealed. Adjusting the direction of processing can reduce the number of non-faithful transitions in the transducers.

The following sections describe these two techniques in detail, and present empirical results on how they affect both the structure of the learned transducers and the size of the characteristic samples in four case studies. These case studies are the same as those analyzed in the previous chapter, allowing for a direct comparison of how the new techniques influence learning outcomes.

4.1 Learning k-value for individual features

Previous literature shows the importance of the properties of k-ISL functions for modeling and learning morpho-phonological processes (Chandlee, 2014; Chandlee et al., 2018, 2014). The notion of Strict Locality allows significant reduction in the memory to represent a particular phenomenon. However, under standard segmental representations, a fixed k value is imposed on the entire segment, even if only some of its subcomponents are affected by the process. In this section, we show that factoring segments by features enables further reduction in memory and characteristic sample, and that the k value can be successfully learned individually for each feature.

A key observation is that only those features which undergo some alternations need a k window greater than 1. All other features, which surface values are identical to their underlying forms, can be modeled with a 1-ISL DFT consisting of a single state, to the exclusion of λ, $\rtimes$, and $\ltimes$ states. Thus, under an input-output identity assumption, faithful features can be learned even in the absence of a characteristic sample.

Informally, this can be illustrated with the Yowlumne process introduced in Chapter 3, where underlying long vowels shorten when followed by two consecutive consonants. The feature [long] is the only one affected by the process and thus the only one requiring $k = 3$. All other features $\phi \in F$ remain unchanged, and can be learned with $k = 1$.

Definition 4.1 (Feature locality minimization). *Let F be a finite set of features and $f : \Sigma^* \to \Gamma^*$ a total function. Featural locality minimization is the process of determining, for each $\phi \in F$, the smallest k_ϕ such that, for some feature set $\Phi \subseteq F$, f_ϕ is representable by a k_ϕ-ISL DFT $\tau(\Phi, \phi)$, and the outputs of $\tau(\Phi, \phi)$ can be inferred from a characteristic sample $S(\Phi, \phi)$.*

To put it in other words, the learning procedure is extended to search not only for the relevant feature set Φ needed to predict ϕ, but also for the minimal k for which the mapping is learnable. The procedure begins with $k = 1$ and checks whether the projection $S(\Phi, \phi)$ yields a functional sample. If not, the value of k is incremented by 1 and the process repeats. Crucially, this is done independently for each $\phi \in F$.

Proposition 4.1. *Let $f : \Sigma^* \to \Gamma^*$ be a total function. Suppose that for each feature $\phi \in F$, there exists a finite subset $\Phi \subseteq F$ and a minimal $k_\phi \in \mathbb{N}$ such that f_ϕ is representable by a k_ϕ-ISL DFT $\tau(\Phi, \phi)$. Then, the feature locality minimization procedure is guaranteed to terminate and return, for each $\phi \in F$, a correct k_ϕ-ISL transduction from a characteristic sample.*

The result follows from the assumption that each f_ϕ is k-ISL for some finite k, and from the existence of characteristic samples of bounded size that ensure learnability for k-ISL transductions (Jardine et al., 2014). Since the learner begins with $k = 1$ and increases k only when the projected sample is not functional, it is guaranteed to reach a value k_ϕ for which the corresponding characteristic sample can be constructed, and the output function inferred.

This strategy reflects the broader theoretical claim that locality constraints apply not globally but at the level of individual features. By assigning the minimal k value necessary for each $\phi \in F$, the learner avoids imposing unnecessary memory bound on faithful features and, as a consequence, reduces the total size of required characteristic sample.

Although the definition of featural locality minimization does not impose any upper bound on the size of the memory window k, prior work suggests that only relatively small values are needed (Chandlee, 2014; Chandlee et al., 2015; Jardine, 2016). Based on this empirical boundary, the learning procedure in the experiments that follow restricts the search to $k \in \{1, 2, 3, 4\}$. This practical constraint reflects both typological findings and computational tractability, while still ensuring successful learning.

4.2 Directionality and string processing

Up until this point, the ISL functions were discussed under the assumption that the input string is processed from left to right. That is, the function reads the string from the left boundary symbol ($\rtimes$) to the right boundary symbol ($\ltimes$), producing output along the way. Importantly, however, the structure of a k-ISL DFT is based entirely on the k-long substrings of the input. When dealing with total functions, where the transducer includes transitions for all possible k-long input contexts, the direction of processing, left to right (L2R) or right to left (R2L), does not affect the underlying shape of the machine. The only difference lies in the outputs assigned to transitions, particularly those that reflect some alternations, i.e. are *non-identical* to the corresponding inputs.

In morpho-phonology, directionality plays a critical role in shaping the relationship between the input and the context that triggers a transformation. Rules depend on the left-hand context, such as back vowel harmony in Turkish, right-hand context, such as vowel shortening in Yowlumne, and some require context on both sides, as in English flapping. For instance, a rule of the form $C \rightarrow D \ / \ \_\_ \ A$ says that C undergoes a change to D if it is followed by A. If processing proceeds from left to right, the learner encounters C before seeing the relevant context. In such cases, the transducer must delay the decision, here by outputting the empty string λ, until the trigger is observed. As a result, the transducer will include three types of transitions: identity transition, where the output matches the input, λ-transitions representing delayed

output or deletion, and non-identity transitions, where the alternation is presented overtly.

In contrast, if the same function is processed from right to left, the relevant context is encountered before the target. Returning to the example from previous paragraph, an R2L learner would see the context A first, allowing it to immediately determine the output D upon reading the target C. In this configuration, no delay is required; the transformation can be accomplished while in the state representing the context. As such, right-to-left processing can eliminate the need for λ-transitions in cases involving right-hand context, potentially reducing the number of non-faithful transitions required to model the process.

Some morpho-phonological processes appear to require context on both sides of the target. In such cases, the number of non-faithful transitions may remain constant across directions, especially when the context includes exactly one element on each side. Since both must be observed to determine the output, λ-transitions are unavoidable. The English flapping rule is a classic example: /t/ becomes [ɾ] when surrounded by vowels (here, the role of stress is omitted for clarity). At first glance, the Chukchi process described in Chapter 3 may appear similar. However, due to its different nature, specifically that it involves insertion of a symbol and not the change of the symbol, and its dependence on the right word boundary, the effective context is in fact exclusively right-sided: the learner must identify two consonants at the end of a word, immediately before $\ltimes$. As a result, right-to-left processing again reduces the number of non-faithful transitions. The functions analyzed in this chapter are the same as in the previous section and do not include cases of truly bidirectional context.

This observation leads to the hypothesis that aligning the direction of processing with the location of the triggering context can yield transducers with fewer transitions to learn. This, in turn, suggests that fewer required transitions may reduce the size of the characteristic sample needed for successful learning. Returning once again to the running example, under left-to-right processing, the learner must see data such

as CAD $\rightarrow$ DAD, CAC $\rightarrow$ DAC, CA $\rightarrow$ CA, and so on. In contrast, under right-to-left processing, the learner could, in principle, succeed with a single representative alternation (e.g., CAD $\rightarrow$ DAD), while the remaining transitions would be correctly inferred as identity mappings.

To evaluate this hypothesis, each of the functions described in Section 3.4 was re-learned with right-to-left processing, complementing the earlier left-to-right results. This allows for a direct comparison of the number of non-faithful transitions derived under each direction. In addition, each function was learned both with a fixed value of k and with the k-learning procedure for each feature individually, introduced in the previous section.

Taken together, these two extensions, feature locality minimization and the context sensitive selection of processing direction, point to the importance of aligning structural assumptions with the nature of the problem at hand. Rather than imposing uniform locality bounds or a fixed direction of reading the input, the learner tailors its strategy to the behavior of individual features and the location of the trigger. The following section presents empirical results demonstrating how these choices affect the number of non-faithful transitions and consequently the overall complexity of the learned transducer. Importantly, the results also show how the sizes of the characteristic samples vary under different configurations, which demonstrates the direct impact of incorporating flexibility into the learning model.

4.3 Results

This section presents the results of experiments evaluating how the featural learner generalizes k-ISL phonological processes when the locality bound k and the direction of processing (left-to-right vs. right-to-left) are varied. The main objective is to determine how these parameters affect the size of the characteristic sample required for successful generalization. Tables 2.1 through 2.11 summarize the key findings across

the four case studies, focusing on representative sample sizes sufficient to induce generalization for each feature ($M_{(\Phi,\phi)}$), as well as for all features combined (M_F).

In addition to characteristic sample size, we also discuss how variations in k and processing direction impact the structure of the learned transducers. Specifically,we examine how the number of states and the number of non-faithful transitions change as a function of these parameters. These statistics provide additional insight into the generalization behavior of the learner. Full details regarding the size and structure of the transducers are provided in Tables A.1–A.14 in Appendix A. Finally, after presenting the descriptive results for the four case studies, we statistically evaluate whether processing direction and feature-specific k learning reliably predict reductions in the reduced per-feature sample size $M_{(\Phi,\phi)}$.

4.3.1 English

The results for learning vowel nasalization in English demonstrate that both processing direction and feature-specific locality minimization significantly impact the size of individual characteristic samples. As shown in Table 4.1 on the left, where k was fixed at 2 for all features, the average sample sizes under right-to-left (R2L) processing are only modestly smaller than those under left-to-right (L2R). However, targeted comparisons reveal more substantial differences. For instance, in the best-case scenario, SOSFIA successfully generalized the [nasal] feature using just 9 input–output pairs in the R2L condition, whereas the smallest L2R sample required 24 pairs to achieve the same result.

These differences between direction of processing disappear when k is learned separately for each feature (Table 4.1, right side). This pattern is expected, particularly for features that do not exhibit any alternations. In all those cases the function was inferred for with the smallest k. Moreover, binary features were learned from only 3 examples, and ternary features from 4. Following Proposition 1 SOSFIA requires only

Feature set Φ	**Fixed** $k = 2$				**Learned** k				k
	L2R		R2L		L2R		R2L		
	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD	
[**voice**]	9	2.66	8	3.91	2	0	2	0	1
[**cons**]	9	2.02	5	2.13	2	0	2	0	1
[**cor**]	23	4.76	16	11.17	3	0	3	0	1
[**lat**]	28	4.06	18	11.77	3	0	3	0	1
[**nasal**, cons]	31	3.89	29	10.10	32	4.32	32	3.90	2

TABLE 4.1: Comparison of average sample sizes $|\mathbf{M}_{(\Phi,\,\phi)}|$ and standard deviations (SD) for VN in English under left-to-right (L2R) and right-to-left (R2L) processing. The table contrasts results for fixed $k = 2$ and learned k across all features.

strings of length one to compute the correct generalization when $k = 1$, which in this case indicates that all available examples were used. As predicted, the k value for the [nasal] feature could not be reduced to 1, and as a result, the average characteristic sample sizes for that feature were comparable to those observed in the fixed-k condition.

$|\mathbf{S}|$	**Segments**		**Features**		**Direction**	k
	Avg. $|\mathbf{M}|$	SD	Avg. $|\mathbf{M}_F|$	SD		
18,278	6,157	650.21	188	10.06	L2R	2
			137	15.87	R2L	2
			46	3.71	L2R	Learned k
			44	1.89	R2L	Learned k

TABLE 4.2: Averaged segmental samples $|\mathbf{M}|$ and synthesized featural $|\mathbf{M}_\mathrm{F}|$ with standard deviations (SDs) in English for the four conditions.

Table 4.2 summarizes the size of synthesized samples M_F sufficient to generalize all features $\phi \in F$ together. With a fixed $k = 2$, the learner succeeded in learning the entire featural function using an average of 137 examples in the R2L setting, with the smallest round requiring only 114. In contrast, the L2R direction resulted in consistently larger samples, with an average of 188 and a minimum of 177 input–output pairs. When k was inferred independently for each feature, the total sample size dropped by more

than 50%, requiring only 46 (L2R) and 44 (R2L) examples on average. These results clearly highlight the effectiveness of the featural learner, both in terms of efficiency and representational flexibility. The changed direction of processing and the ability to tailor k for each individual feature allow the learner to exploit structural properties of the data, leading to substantially smaller characteristic samples. Detailed results from all ten runs across the four experimental settings are provided in Appendix B (Tables B.1–B.8).

4.3.2 Yowlumne

The results for Yowlumne vowel shortening reveal similar patterns, with even more pronounced advantages of the new parameters. With fixed $k = 3$ (Table 4.3, left), reversing direction led to a substantial reduction for the [long] feature, from an average of 317 examples in L2R to just 169 in R2L. While [ant] unexpectedly showed larger R2L $M_{(ant,ant)}$ samples (avg. 195 vs. 136), individual rounds show this is likely due to the random sampling type. For example, the smallest R2L sample had just 4 examples, while L2R required at least 104 (see detailed results in Appendix B, Tables B.13 - B.20). Since AMßA selects examples randomly in batches of 10, the larger R2L average likely reflects unlucky sampling rather than a systematic difference.

When k was learned per feature (Table 4.3, right), all features except [long] required only 3–4 examples, similarly to English results, and identified $k = 1$ where appropriate. For feature [long], the inferred $k = 3$ was retained, but the average sample size still dropped from 250 in L2R to 155 in R2L.

Table 4.4 shows the synthesized sample sizes $|\mathbf{M}_F|$ across all features. Compared to the segmental baseline (avg. $|M| = 16,583$), the featural learner required just 952 (L2R) and 851 (R2L) examples with fixed $k = 3$. When k was inferred separately, $|\mathbf{M}_F|$ samples dropped even further to 258 and 164, respectively. This reflects an 82% reduction in required data relative to the fixed-k L2R default settings. This case study

Feature set Φ	**Fixed** $k = 3$				**Learned** k				k
	L2R		R2L		L2R		R2L		
	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD	
[**voice**]	33	10.11	32	5.21	2	0	2	0	1
[**round**]	30	7.39	35	7.21	2	0	2	0	1
[**ant**]	136	19.28	195	74.54	3	0	3	0	1
[**cor**]	139	24.05	174	46.11	3	0	3	3	1
[**long**]	317	49.02	169	17.58	250	63.26	155	21.02	3

TABLE 4.3: Comparison of average sample sizes $|\mathrm{M}_{(\Phi,\,\phi)}|$ and standard deviations (SD) for vowel shortening (VS) in Yawelmani under left-to-right (L2R) and right-to-left (R2L) processing. The table contrasts results for fixed $k = 3$ and learned k across all features.

$\lvert\mathbf{S}\rvert$	**Segments** Avg. $\lvert\mathbf{M}\rvert$	SD	**Features** Avg. $\lvert\mathbf{M}_F\rvert$	SD	**Direction**	k
17,206	16,583	702.06	952	49.90	L2R	3
			851	56.51	R2L	3
			258	62.95	L2R	Learned k
			164	21.24	R2L	Learned k

TABLE 4.4: Averaged segmental samples $|\mathbf{M}|$ and synthesized featural $|\mathrm{M_F}|$ with standard deviations (SDs) in Yawelmani for the four conditions.

demonstrates even more clearly that the flexible feature-based model enables highly efficient learning of ISL functions from dramatically smaller characteristic samples.

4.3.3 Chukchi

The Chukchi case presented different results from the two previous cases, likely due to the nature of the process being modeled: insertion vs. change in feature value. As shown in Table 4.5, reversing processing direction had little to no impact on the average size $M_{(\Phi,\phi)}$ samples for individual features (e.g. [round]). Features [labial] and [continuant] revealed smaller averaged samples (152 in L2R vs. 313 in R2L condition for ϕ =labial). Conversely, several other features such as [anterior] or [coronal] even showed higher R2L sample means. These results diverge from the strong directional effects observed in earlier case studies. More importantly, no features, both sufficient and insufficient, were able to reduce the locality bound k below 3. This outcome is unsurprising given that segmental epenthesis affects all features simultaneously.

	L2R		R2L	
Feature set Φ	Avg. $\lvert M_{(\Phi,\phi)}\rvert$	SD	Avg. $\lvert M_{(\Phi,\phi)}\rvert$	SD
[**tense**]	32	5.69	36	9.56
[**high**]	165	30.34	192	51.10
[**round**, high]	225	30.28	227	35.57
[**lab**]	313	8.88	152	24.53
[**cont**, tense]	306	18.02	219	31.15
[**strid**, dor]	955	58.46	964	46.30

TABLE 4.5: Selected representative samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for FE in Chukchi with left-to-right (L2R) and right-to-left (R2L) processing for set $k = 3$.

Similarly, synthesized sample sizes $\lvert M_F\rvert$ remained fairly comparable across directions (Table 4.7), with averages of 2,331 for L2R and 2,352 for R2L. This again contrasts with English and Yowlumne case studies, where both directionality and feature-specific k optimization led to substantial reductions in required data. The lack of variability here may in part reflect the randomized nature of AMßA sampling and the

Feature set Φ	Fixed $k = 3$				Learned k				k
	L2R		R2L		L2R		R2L		
	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD	
[**tense**]	32	5.69	36	9.56	35	6.40	43	4.33	3
[**high**]	165	30.34	192	51.10	165	24.44	173	37.57	3
[$\substack{\textbf{round}\\\text{high}}$]	225	30.28	227	35.57	225	38.34	216	39.18	3
[**lab**]	313	8.88	152	24.53	302	20.73	151	34.33	3
[$\substack{\textbf{cont}\\\text{tense}}$]	306	18.02	219	31.15	300	14.02	226	29.76	3
[$\substack{\textbf{strid}\\\text{dor}}$]	955	56.46	964	46.30	957	103.34	991	90.65	3

TABLE 4.6: Comparison of average sample sizes $|\mathbf{M}_{(\Phi,\,\phi)}|$ and standard deviations (SD) for FE in Chukchi under left-to-right (L2R) and right-to-left (R2L) processing. The table contrasts results for fixed $k = 3$ and learned k across all features.

nature of the process.

$\lvert\mathbf{S}\rvert$	Segments		Features		Direction	k
	Avg. $\lvert\mathbf{M}\rvert$	SD	Avg. $\lvert\mathbf{M}_F\rvert$	SD		
17,206	17,181	28.87	2,331	46.27	L2R	3
			2,352	43.24	R2L	3
			2,338	101.13	L2R	Learned k
			2,412	75.47	R2L	Learned k

TABLE 4.7: Averaged segmental sample $|M|$ and averaged synthesized featural $|M|_{\mathrm{F}}$ and their corresponding SDs in Chukchi for the two conditions.

4.3.4 Polish

The results for Polish opaque interaction of final devoicing and ɔ-raising show a mixed effect of processing direction and feature-specific k learning on both the averaged individual samples $M_{(\Phi,\phi)}$ and the synthesized samples M_F (Table 4.9). As shown in Table 4.8 (left), with a fixed $k = 3$, reversing the processing direction produced no consistent overall change in average sample sizes. In some cases, R2L setting yielded slightly smaller samples (e.g. for ϕ =[tense]), but in others, the averages were slightly larger than in L2R (e.g. for ϕ =[voice]). For [voice] in particular, the average $|M_{([\substack{\textbf{voice}\\\text{son}}],\text{voice})}|$

in R2L was roughly 20 examples larger than in L2R. However, a closer look at the ten individual runs (Tables B.37 and B.38 in Appendix B) reveals that reversing direction can, in some cases, lead to very small samples: the smallest R2L run for [voice] required only 4 examples, whereas the smallest L2R run required 196.

Feature set Φ	**Fixed** $k = 3$				**Learned** k				k
	L2R		R2L		L2R		R2L		
	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD	
[**voice** / son]	214	10.73	233	85.93	193	29.18	101	38.13	2
[**nasal**]	29	14.92	28	10.19	2	0	2	0	
[**son**]	27	12.77	28	10.13	2	0	2	0	1
[**round**]	137	51.76	173	139.89	3	0	3	0	1
[**high** / voice / son / low]	7119	907.70	7360	630.27	6984	1262.72	7156	695.43	3
[**tense** / voice / son / round]	7372	806.83	7052	597.12	7176	776.65	7240	578.20	3

TABLE 4.8: Comparison of average sample sizes $\lvert\mathbf{M}_{(\Phi,\,\phi)}\rvert$ and standard deviations (SD) for FD and OR in Polish under left-to-right (L2R, $\longrightarrow$) and right-to-left (R2L, $\longleftarrow$) processing. The table contrasts results for fixed $k = 3$ and learned k across all features.

When k was learned individually for each feature (Table 4.8, right), the effects of feature locality minimization became more apparent. For features affected by final devoicing, the window size was reduced to $k = 2$, and sample sizes dropped substantially, often staying below 80 examples in R2L (Tables B.39 and B.40 in Appendix B). This is particularly interesting in the context of process interaction. Here, final devoicing and ɔ-raising were distinguished automatically by the procedure, with the former requiring a smaller locality while the latter remained $k = 3$. This is important for learning morphophonological grammar, as in realistic settings learners (e.g. children) are exposed to input containing multiple interacting (and non-interacting) processes. Generalization over features can help to separate these processes and may even reveal distinctions between processes that have previously been analyzed as a single phenomenon.

|S|	Segments		Features		Direction	k
	Avg. |M|	SD	Avg. $\|M_F\|$	SD		
17,206	15,169	2,063.50	8,235	208.39	L2R	3
			8,211	139.89	R2L	3
			8,114	145.28	L2R	Learned k
			8,069	202.83	R2L	Learned k

TABLE 4.9: Averaged segmental samples $|\mathbf{M}|$ and synthesized featural $|\mathbf{M}_\mathrm{F}|$ with standard deviations (SDs) in Polish for the four conditions.

In contrast, features directly involved in ɔ-raising, such as [high] and [tense], showed no substantial benefit from either processing direction or feature-specific k-learning. Their average sample sizes remained large in all four conditions, exceeding 7,000 input-output pairs. The $M_{(\Phi,\phi)}$ samples for features not affected in either process, i.e. *sufficient* features, such as [nasal], [son], and [round], were consistently reduced to 2–3 examples in both directions, in the learned k condition.

As in the previous chapter, we also examined the effect of growing alphabets. The results (Tables B.45–B.48 in Appendix B) show no additional reductions beyond those already observed for features affected by final devoicing and for faithful features when $|\Sigma| = 7$. This pattern held as the alphabet size increased. Additionally, sample sizes for [high] and [tense] grew for $|\Sigma| = 9$ and $|\Sigma| = 10$, which was expected as bigger segmental inventory also caused $\Sigma_{[high]}$ and $\Sigma_{[tense]}$ to grow.

Overall, the results for Polish show only modest effects of feature-specific k minimization and reversing the direction of processing. Before turning to the structural explanation for these patterns in the Discussion section, the following subsection evaluates statistically whether processing direction and feature-specific k learning reliably predict reductions in the reduced per-feature sample size $M_{(\Phi,\phi)}$ across the four case studies.

4.3.5 Statistical evaluation of directionality and learned k

The descriptive results reported above suggest that the effect of processing direction and feature-specific k learning differs across the four case studies. To evaluate these patterns statistically, we fit negative-binomial mixed-effects models with the reduced per feature sample size $M_{(\Phi,\phi)}$ as the dependent variable. Statistical evaluations were carried out in R using the `lme4` package, specifically the `glmer.nb()` function. For each individual rule, the model included fixed effects of processing direction (L2R vs. R2L), k condition (fixed vs. learned), and their interaction, with random effects by feature set. Since the same feature sets were tested across conditions, this structure accounts for the possibility that observations from the same feature set are not independent, and that different feature sets may have different expected $M_{(\Phi,\phi)}$ sample sizes or respond differently to changes in processing direction or k condition.

We also fit a global model over all four rules, pooling the feature-level sample-size results across all features, feature sets, directions, and k conditions. This model included random intercepts for feature set and rule, which allows expected reduced sample sizes to vary across feature sets and across rules. This accounts for the fact that some feature sets and some rules may generally require larger or smaller $M_{(\Phi,\phi)}$ samples, independently of the effects of processing direction and k condition. In all models, the L2R fixed-k condition was used as the reference condition.

Since the models use a negative-binomial distribution with a log link, the coefficients are reported on the log scale. The coefficient β indicates the direction and size of the effect on the expected reduced sample size $M_{(\Phi,\phi)}$: negative values indicate smaller expected samples, and positive values indicate larger expected samples. The standard error (SE) measures the uncertainty around the estimated coefficient, with smaller SE values indicating more precise estimates. The p-value indicates whether there is statistical evidence that the effect differs from zero, with $p < .05$ treated as significant.

To make the coefficients easier to interpret, we also report incidence-rate ratios

(IRRs), which are obtained by exponentiating the coefficients. An IRR below 1 means that the expected reduced sample size is smaller than in the reference condition, while an IRR above 1 means that it is larger. For example, an IRR of 0.70 means that the expected sample size is about 70% of the reference condition, or about 30% smaller. With the L2R fixed-k condition as the reference condition, the coefficient for R2L estimates the effect of changing direction while keeping k fixed. The coefficient for learned k estimates the effect of learning k separately for each feature under L2R processing. The interaction term tests whether the effect of learned k differs under R2L processing. The results are summarized in Table 4.10 below.

Rule	Effect	β	SE	p	IRR
English	Direction: R2L	-0.329	0.041	$< .001$	0.720
	Learned k	-1.859	0.113	$< .001$	0.156
	R2L $\times$ learned k	0.328	0.076	$< .001$	1.388
Yowlumne	Direction: R2L	-0.048	0.037	.191	0.953
	Learned k	-3.097	0.175	$< .001$	0.045
	R2L $\times$ learned k	0.036	0.062	.557	1.037
Chukchi	Direction: R2L	-0.054	0.058	.352	0.947
	Learned k	-0.006	0.014	.687	0.994
	R2L $\times$ learned k	0.022	0.020	.266	1.022
Polish	Direction: R2L	0.053	0.042	.198	1.055
	Learned k	-2.492	0.303	$< .001$	0.083
	R2L $\times$ learned k	-0.140	0.072	.053	0.870
Global	Direction: R2L	-0.081	0.043	.057	0.922
	Learned k	-1.743	0.049	$< .001$	0.175
	R2L $\times$ learned k	-0.020	0.062	.751	0.981

TABLE 4.10: Negative-binomial mixed-effects model results for the effects of processing direction, k condition, and their interaction on reduced sample size.

For English vowel nasalization, changing the processing direction significantly reduced the expected reduced sample size in the fixed-k condition, and learning k separately for each feature significantly reduced the sample size under L2R processing. The interaction was also significant, but it was positive. This means that the benefit of R2L processing was much smaller in the learned-k condition than in the fixed-k condition. Put it differently, once k is learned separately for each feature, changing direction

does not provide an additional reduction of the same size.

For Yowlumne vowel shortening, only the effect of learning k was significant. The effect of changing direction was not significant, and neither was the interaction between direction and k condition. This indicates that the main statistically supported reduction for Yowlumne comes from feature-specific locality minimization, rather than from reversing the direction of processing.

For Chukchi epenthesis, none of the tested effects were significant. This is consistent with the descriptive results reported above: neither changing direction nor learning k separately produced a reliable reduction in the sample size. This result is also not surprising given the nature of the process. Since epenthesis affects all features simultaneously, feature-specific k learning cannot isolate a small set of alternating features and reduce the remaining features to simple identity mappings. Although R2L processing reduces the number of non-faithful transitions in the corresponding transducers, this structural reduction does not translate into a statistically significant reduction in sample size.

For Polish, the model showed a significant effect of learning k, but no significant main effect of direction. The interaction between direction and learned k was marginal, but did not reach the significance threshold adopted here. This suggests a possible tendency for learned k to be somewhat more beneficial in the R2L condition, but the evidence is not strong enough to treat this as a reliable interaction. These results are consistent with the descriptive results: feature-specific k learning helps especially for features involved in final devoicing and for faithful features, but the large machines associated with ɔ-raising, especially [high] and [tense], limit the overall reduction in sample size.

The global model confirms the main cross-language pattern. Across all four rules, learning k separately for each feature had a significant effect on expected reduced sample size. In contrast, the global effect of processing direction was only marginal, and the global interaction between direction and learned k was not significant. Thus,

feature-specific k learning is the most robust factor in these experiments with FxF(SOSFIA), while the effect of directionality is more limited and process-dependent.

Taken together, these statistical results sharpen the descriptive pattern reported above. Feature-specific k learning emerges as the more reliable source of sample-size reduction, whereas the effect of processing direction is weaker and more process-dependent. The hypotheses motivating this chapter are therefore supported only in part: learned k significantly reduces $M_{(\Phi,\phi)}$ in English, Yowlumne, and Polish, and remains significant in the global model, while directionality is significant only for English and marginal in the global model.

However, the models do not by themselves explain why the same manipulations produce large reductions for English and Yowlumne, more limited reductions for Polish, and no reliable reduction for Chukchi. To explain this asymmetry, the Discussion turns to the structure of the learned feature-level transducers, focusing especially on the number of states, the number of non-faithful transitions, and the distribution of identical and non-identical input-output pairs in the characteristic samples.

4.4 Discussion

To understand why processing direction and feature-specific k learning produce statistically reliable reductions in some cases but little or no change in others, it is necessary to look beyond the statistical results and examine the structural properties of the feature-level machines themselves. In particular, the number of states, the number of non-faithful transitions, and the distribution of identical and non-identical pairs help explain why some reductions in transducer complexity translate into smaller characteristic samples, while others do not.

For English and Yowlumne (Table 4.11), reversing the processing direction from L2R to R2L with a fixed k substantially reduced the number of non-faithful transitions: from 6 to 1 in English (–83%) and from 24 to 1 in Yowlumne (–98%). These reductions

were reflected in the average synthesized sample sizes M_F, which decreased by 27% in English and 11% in Yowlumne.

Condition	**Feature ϕ**	$\|\mathbf{Q}\|$	$\|\mathbf{\Sigma_\Phi}\|$	$\|\delta\|$	$\neg$ **faith.** $\|\delta\|$	$\|\mathbf{S}_{(\Phi,\phi)}\|$	$\neg$ **faith.** $\|\mathbf{S}_{(\Phi,\phi)}\|$	**k**
				English				
set k, L2R	[nasal]	6	3	17	6	39	7	2
	[cor]	6	3	17	0	39	0	2
(avg. $\|M_F\| = 188$)	**SUM**	126	60	332	6	701	7	
set k, R2L	[nasal]	6	3	17	1	39	7	2
	[cor]	6	3	17	0	39	0	2
(avg. $\|M_F\| = 137$)	**SUM**	126	60	332	1	701	7	
learned k, L2R	[nasal]	6	3	17	6	39	7	2
	[cor]	3	3	5	0	3	0	1
(avg. $\|M_F\| = 46$)	**SUM**	69	60	116	6	96	7	
learned k, R2L	[nasal]	6	3	17	1	39	7	2
	[cor]	3	3	5	0	3	0	1
(avg. $\|M_F\| = 44$)	**SUM**	69	60	116	1	96	7	
				Yawelmani				
set k, L2R	[long]	15	3	53	24	363	34	3
	[cor]	15	3	53	0	336	0	3
(avg. $\|M_F\| = 952$)	**SUM**	270	57	807	24	4,028	34	
set k, R2L	[long]	15	3	53	1	363	34	3
	[cor]	15	3	53	0	336	0	3
(avg. $\|M_F\| = 851$)	**SUM**	270	57	807	1	4,028	34	
learned k, L2R	[long]	15	3	53	24	363	34	3
	[cor]	3	3	5	0	3	0	1
(avg. $\|M_F\| = 258$)	**SUM**	84	57	153	24	417	34	
learned k, R2L	[long]	15	3	53	1	363	34	3
	[cor]	3	3	5	0	3	0	1
(avg. $\|M_F\| = 164$)	**SUM**	85	57	153	1	417	34	

TABLE 4.11
FxF(SOSFIA) complexity and sample-size statistics for English and Yawelmani under fixed vs. learned k and L2R vs. R2L processing. Rows show selected output features and the corresponding columns size ($|Q|$, $|\Sigma_\Phi|$, $|\delta|$), non-faithful transitions, and characteristic sample size $|S_{(\Phi,\phi)}|$; **SUM** aggregates over all features.

An even larger effect came from learning k separately for each feature. While some features (e.g., [nasal] in English and [long] in Yowlumne) remained unchanged, others showed major reductions in state space—for example, [coronal] in English dropped

from 6 to 3 states, and in Yowlumne from 15 to 3 states. This corresponded to reductions in the number of transitions: from 17 to 5 (–71%) for [coronal] in English and from 53 to 5 (–91%) for [coronal] in Yowlumne. Summing the number of transitions across all features, the number fell by 65% in English and 81% in Yowlumne with the minimized k values. Consequently, the sum of the characteristic sample sizes $S_{(\Phi,\phi)}$ across features dropped by 86% in English and 90% in Yowlumne, yielding synthesized M_F reductions of 76% and 73% , respectively.

Applying both conditions, per-feature k learning and reversing direction, produced the largest reductions in sample sizes: 77% for English and 83% for Yowlumne.

For Chukchi (Table 4.12), [strident] required the largest machine among all features. Because every feature in this system undergoes changes, reducing k was impossible for any of them. Reversing the processing direction did substantially reduce the number of non-faithful transitions (e.g., [strident]: 54 to 9, –83%; [round]: 6 to 1, –83%), similar to the scale of reductions in English and Yowlumne. However, these reductions did not translate into proportional decreases in the overall sample size.

A key difference lies in the distribution of identical vs. non-identical pairs. In English, the ratio of identical to non-identical pairs for the affected features was about 5:1; in Yowlumne, it was roughly 10:1. Considering all features together, the ratios were 99:1 (English) and 117:1 (Yowlumne). In Chukchi, by contrast, some features like [strident] had almost equal numbers of identical and non-identical pairs, while others (e.g., [round]) had ratios more similar to English. Averaged over all features, the ratio was close to 1:1. That means that, on average, each feature required to see at least half of its original characteristic sample to assure successful inference.

It is also worth noting that the sum of non-faithful pairs across features (4,007) is not directly comparable to the number of examples needed to learn all features together (approximately 2,300), because the sum of characteristic samples $S_{(\Phi,\phi])}$ for all $\phi \in F$ counts each feature's sample separately, whereas M_F is a single segmental sample whose members can serve multiple features simultaneously. This helps explain

Condition	**Feature ϕ**	**$\lvert Q \rvert$**	**$\lvert \Sigma_\Phi \rvert$**	**$\lvert \delta \rvert$**	**$\neg$ faith. $\lvert \delta \rvert$**	**$\lvert S_{(\Phi,\phi)} \rvert$**	**$\neg$ faith. $\lvert S_{(\Phi,\phi)} \rvert$**	**k**
				Chukchi				
set k, L2R	[strid]	23	4	106	54	1,236	669	3
	[round]	15	3	53	6	309	40	3
(avg. $\lvert M_F \rvert = 2,331$)	**SUM**	320	63	1,161	755	9,074	4,007	
set k, R2L	[strid]	23	4	106	9	1,236	669	3
	[round]	15	3	53	1	309	40	3
(avg. $\lvert M_F \rvert = 2,352$)	**SUM**	320	63	1,161	75	9,074	4,007	
learned k, L2R	[strid]	23	4	106	54	1,236	669	3
	[round]	15	3	53	6	309	40	3
(avg. $\lvert M_F \rvert = 2,338$)	**SUM**	320	63	1,161	755	9,074	4,007	
learned k, R2L	[strid]	23	4	106	9	1,236	669	3
	[round]	15	3	53	1	309	40	3
(avg. $\lvert M_F \rvert = 2,412$)	**SUM**	320	63	1,161	75	9,074	4,007	
				Polish				
set k, L2R	[high]	45	6	302	143	8,250	1,375	3
	[voice]	15	3	53	21	336	112	3
	[ant]	15	3	53	0	336	112	3
(avg. $\lvert M_F \rvert = 8,235$)	**SUM**	255	47	1,089	307	18,792	2,862	
set k, R2L	[high]	45	6	302	2	8,250	1,375	3
	[voice]	15	3	53	1	336	112	3
	[ant]	15	3	53	0	336	112	3
(avg. $\lvert M_F \rvert = 8,211$)	**SUM**	255	47	1,089	5	18,792	2,862	
learned k, L2R	[high]	45	6	302	143	8,250	1,375	3
	[voice]	6	3	17	6	39	13	2
	[ant]	3	3	5	0	3	0	1
(avg. $\lvert M_F \rvert = 8,114$)	**SUM**	138	47	681	292	16,571	2,763	
learned k, R2L	[high]	45	6	302	2	8,250	1,375	3
	[voice]	6	3	17	1	39	13	2
	[ant]	3	3	5	0	3	0	1
(avg. $\lvert M_F \rvert = 8,069$)	**SUM**	138	47	681	5	16,571	2,763	

TABLE 4.12
FxF(SOSFIA) complexity and sample-size statistics for Chukchi and Polish under fixed vs. learned k and L2R vs. R2L processing. Rows show selected output features and the corresponding columns size ($|Q|$, $|\Sigma_\Phi|$, $|\delta|$), non-faithful transitions, and characteristic sample size $|S_{(\Phi,\phi)}|$; **SUM** aggregates over all features.

why Chukchi's $|M_F|$ did not drop as significantly as in English or Yowlumne.

The Polish case (Table 4.12) shows a similar pattern. Ratios of identical to non-identical pairs in $S_{(\Phi,\phi])}$ ranged from about 5:1 for [high] (affected by ɔ-raising) to about

3:1 for [voice] (affected by final devoicing). For sufficient features, the ratios were around 6:1 with fixed $k = 3$ and 5:1 in learned k condition, which is similar to English. Nevertheless, M_F samples do not show any substantial reductions in learning Polish processes. This indicates that this ratio alone cannot explain the lack of large sample-size reductions.

A major factor is that the insufficient features [high] and [tense] remained almost as large as their original segmental machines. With $|\Sigma| = 7$, the alphabets Σ_{high} and Σ_{tense} were reduced by only 1 after feature factorization. In L2R direction, the machine for feature [high] had 45 states and 143 non-faithful transitions to learn. In contrast, the largest machine in the previous experiments had 23 states and 54 non-faithful transitions. While reversing direction reduced the number of transitions with non-faithful outputs to 2, the probability of sampling the particular two input-output pairs covering those transitions was extremely low given the total sample size ($|S_{(\left[\begin{smallmatrix}\textbf{tense}\\ \text{voice}\\ \text{son}\\ \text{round}\end{smallmatrix}\right], tense)}| = 8,250$) and the batch size of 10 used for this feature.

Finally, even when identical outputs are assumed at the start of learning, the way SOSFIA calculates the outputs for each transition can influence the outcome. Having all the relevant non-identical input-output pairs in the characteristic sample is not always sufficient. What also matters is what identical pairs are included. In some cases, SOSFIA's `lcp` procedure can assign outputs earlier than expected which results in machine that diverges from the original. For example, consider a simplified process where D changes to T word-initially (D $\longrightarrow$ T / $\rtimes$ __, with $\Sigma = \{D, V\}$, where $V = \overline{D}$, and a characteristic sample S = {(D,T), (DV,TV), (VVV,VVV)}. The essential change from D to T is learned from the first two pairs. However, when learning the transition $\delta = (\lambda, V, V, V)$ SOSFIA sees only one pair, (VVV,VVV), whose output has an `lcp` of VVV, and rewrites the transition as $\delta = (\lambda, V, VVV, V)$. Such outcomes illustrate that, SOSFIA fails to generalize when critical datapoints are missing from the sample.

From the results and analyses presented in this chapter, it becomes clear that the

inconsistent effects of per-feature k learning and processing direction are not simply a matter of these procedures working for some processes and failing for others. Instead, their impact is shaped by a combination of other factors: the sampling procedure, the structural properties of the underlying machines, and the nature of the data generated for the experiments. Importantly, the behavior of the chosen learner itself seems to play a significant role, as the algorithm is not always well suited to making full use of the characteristic samples and in some cases it may fail to exploit even seemingly unimportant data. These considerations help explain why some processes show notable improvements under the additional procedures, while others display little to no change.

In the next chapter, we introduce a more flexible learner that determines its own k value, processing direction, and relevant tiers. The procedure for calculating outputs will also differ from SOSFIA's, addressing one of the limitations identified in the present chapter. The experiments in the next chapter focus on the best-performing settings identified here, rather than testing all combinations of direction and k condition as was done for SOSFIA. The new learner will also be tested on datasets drawn from multiple phonology textbooks. Because these datasets are considerably smaller, they present conditions under which SOSFIA would be expected to fail.

Chapter 5

New Learner: SEAL

The previous chapter showed that learning an individual k value for each feature, as well as manipulating the direction of string processing, is not only possible for k-ISL learning, but also affects learnability outcomes. In particular, we showed that reducing the memory window directly affects the size of the transducer, and also reduces the size of the extracted samples for particular features, as predicted in Chapter 3. Directionality, by contrast, primarily affects the number of non-faithful transitions. Specifically, it can eliminate λ transitions when the processing direction allows the learner to observe the conditioning environment before encountering the target of the change.

The strongest evidence for the impact of these two parameters comes from processes involving a single feature change, particularly in the English and Yowlumne cases. Directionality did not affect epenthesis, since the insertion of ə in Chukchi requires contextual information from both sides of the target, rather than exclusively from the left or the right. In Polish, directionality substantially reduced the number of non-faithful transitions: while the L2R condition with $k = 3$ produced 307 non-faithful transitions all together, the R2L condition yielded only 5. However, this improvement surprisingly did not allow AMßA to find smaller samples for features [high] and [tense].

Together, these results suggest that although directionality can reduce the number of non-faithful transitions and reducing k yields smaller transducers, these parameters alone do not guarantee improved sample efficiency across features. This motivates

the development of a new learner that determines transition outputs in a different way—one that makes more effective use of the information available in the sample. In this chapter, we introduce the String Equation Adaptive Learner (SEAL) and evaluate it on data from English, Yowlumne, Chukchi, and Polish.

5.1 DFT inference via string-equation

While both SOSFIA and SEAL are set to solve the same problem, they offer different solutions. SOSFIA predicts the output on each transition using the longest common prefix of observed outputs for compatible prefixes. SEAL, by contrast, formulates transducer learning as a problem of *string–equation inference*. The central idea is to encode input paths of a subsequential transducer as strings over a new alphabet of placeholders $\{O_i\}_{i\in\mathbb{Z}_{>0}}$, and then infer a globally consistent assignment of output substrings to these placeholders.

5.1.1 Preliminaries

Let Σ and Δ be finite input and output alphabets, respectively, and let $S \subseteq \{\rtimes\}\Sigma^*\{\ltimes\} \times \Delta^*$ be a finite sample of input–output pairs.

SEAL introduces an auxiliary placeholder alphabet $\Omega = \{O_i \mid i \in \mathbb{Z}_{>0}\}$, disjoint from Δ. Placeholders in Ω do not represent output symbols directly; instead, they serve as variables whose values are strings in Δ^*.

Let $\mathcal{T}_\Omega$ be a delimited, onward, subsequential transducer with a unique initial state λ, a designated left boundary state $\rtimes$, and a designated right boundary state $\ltimes$ (with no outgoing transitions). Each transition in $\mathcal{T}_\Omega$ is labeled with an input symbol from $\Sigma \cup \{\rtimes, \ltimes\}$ and an output placeholder from Ω. In this chapter we will be especially interested in the special case where $\mathcal{T}_\Omega$ has the transition structure of a k-ISL DFT, as defined in earlier chapters.

For any delimited input string $\rtimes w \ltimes \in \{\rtimes\}\Sigma^*\{\ltimes\}$, the unique path in $\mathcal{T}_\Omega$ labeled by $\rtimes w \ltimes$ produces an *encoded output string* $w_o \in \Omega^*$ obtained by concatenating the placeholders along that path. Since the initial state λ has exactly one outgoing transition to $\rtimes$, every encoded string begins with the same placeholder on that transition; we denote this placeholder by O_{init}. Likewise, every encoded string ends with the placeholder on the transition into $\ltimes$.

The learner's objective is to infer a mapping $\mu : \Omega \to \Delta^*$ that assigns an output string to each placeholder. This mapping extends naturally to strings in Ω^* by concatenation, s.t. for $w_o = O_{i_1} \cdots O_{i_m} \in \Omega^*$, $\mu(w_o) = \mu(O_{i_1}) \cdots \mu(O_{i_m})$. SEAL succeeds on S if for every $(x, y) \in S$, where $x = \rtimes w \ltimes$, the encoded output w_o produced by $\mathcal{T}_\Omega$ on input $\rtimes w \ltimes$ satisfies $\mu(w_o) = y$. Applying μ to all transitions of $\mathcal{T}_\Omega$ yields a standard DFT $\mathcal{T}_\Delta$ whose transitions are labeled with output strings in Δ^*.

5.1.2 String Equation Adaptive Learner (SEAL)

Given a choice of learning parameters including the memory window k, a set of feature combinations (which determines the projected alphabet), and a processing direction, SEAL first constructs a k-ISL DFT skeleton and assigns a unique placeholder from $\Omega = \{O_i\}_{i \in \mathbb{Z}_{>0}}$ to the output of each transition. The resulting transducer $\mathcal{T}_\Omega$ has the same transition structure as the corresponding k-ISL DFT, but its transition outputs are placeholders rather than strings in Δ^*. Passing each training input x through $\mathcal{T}_\Omega$ yields an encoded placeholder string $w_o \in \Omega^*$.

Learning is then distributing each target output string y (for $(x, y) \in S$) across the placeholders in w_o so that a single global assignment is consistent for all examples. Even for small samples, this induces a large hypothesis space, since each placeholder O_i can admit substrings of various lengths, including the empty string. It is therefore pertinent to develop a strategic but consistent method for assigning substrings to placeholders.

To constrain the search, SEAL imposes an inductive bias grounded in the structure of k-ISL DFTs and the assumption that many phonological mappings are perturbations of identity. The learner therefore begins with the simplest hypothesis: each placeholder contributes at most one output symbol, i.e. an initial 1–to–1 alignment between the encoded string and the target. This initialization is implemented by `Align1to1` (Algorithm 5.1), which aligns the target output y to the encoded sequence position-by-position. When the encoded side is longer than the target (e.g. in deletion and many feature-changing settings), `Align1to1` assigns λ to the trailing placeholders to match length. Boundary placeholders are treated as default-empty. The placeholder on the unique transition into the left boundary state is fixed to λ (denoted $O_{init} = \lambda$), since nonempty output there would predict that every word begins with the same material.[1] The placeholder on the transition to final state ($\ltimes$) allows more flexibility. If the output is longer than the encoded side, `Align1to1` still performs a one-to-one mapping for the first $m - 1$ positions and assigns all remaining output material to the last placeholder.

Algorithm 5.1: `Align1to1`

Data: A sequence $s_1 \cdots s_m$ of length m, an output string $y \in \Delta^*$ tokenized as $y_1 \cdots y_t$ of length t

Result: A list aligned$[1..m]$ with $|\text{aligned}| = m$

if $t = m$ **then**
  return $[y_1, \ldots, y_m]$
else if $t < m$ **then**
  return $[y_1, \ldots, y_t, \lambda, \ldots, \lambda]$
else
  return $[y_1, \ldots, y_{m-1}, (y_m \cdots y_t)]$
end if

Learning proceeds iteratively as described in `PRD-SEAL` (Algorithm 5.2) below.

[1] Although some morphophonological processes may require every output form to be preceded by the same fixed material, SEAL does not assume such cases in the present implementation. Future versions of `Align1to1` could relax this restriction by allowing (O_{init}) to be nonempty when supported by the data.

Algorithm 5.2: Push Right Distribution `PRD-SEAL`

Data: Sample $S \subseteq \{\rtimes\}\Sigma^*\{\ltimes\} \times \Delta^*$, identity hypothesis H_{id}
Result: Updated hypothesis H_{id}

```
R ← [ ]
for all (x, y) ∈ S do
    append (DropLeft(x), Align1to1(DropLeft(x), y)) to R
end for
while Conflicts(R) ≠ ∅ do
    R_prev ← R
    R ← PushRight(R, Conflicts(R))
    if R = R_prev then
        break
    end if
end while
return Merge(H_id, BuildH(R))
```

For each training pair $(x, y) \in S$, the learner first removes the left boundary symbol via `DropLeft` (Definition 5.1) and stores the resulting aligned representation in a working list R. The current partial assignment is then tested for consistency across the dataset. Conflicts are identified by `Conflicts` (Definition 5.2): a placeholder becomes a conflict if it is associated with more than one distinct output token across the rows of R.

Definition 5.1. ***DropLeft*** *For a delimited input string $x = x_1x_2 \cdots x_n$, let $\texttt{DropLeft}(x) = [x_2, \ldots, x_n]$ (i.e. drop only the left boundary symbol).*

Definition 5.2. ***Conflicts*** *Given rows $R = \{(\textit{DropLeft}, \textit{aligned})\}$, a symbol s is a conflict iff it is assigned more than one distinct output token across R. Let $\texttt{Conflicts}(R) = \{\, s \mid |\{t : (s, t) \textit{ appears in some row of } R\}| > 1 \,\}$.*

When such a conflict arises, SEAL resolves it using a *push-right adjustment*, implemented by `PushRight` (Algorithm 5.3). Concretely, whenever a conflicting placeholder occurs at position i, SEAL sets its current output to λ and shifts the displaced material to the immediately following placeholder on the same path, concatenating it with any output already assigned there while preserving order. If the conflict occurs at the final position on the path, the material is retained locally. This operation preserves

previously consistent decisions and restricts changes to the local point of disagreement, while still allowing output material to be redistributed along the placeholder sequence.

Algorithm 5.3: `PushRight`

Data: Rows $R = \{(\text{middle} = s_1 \cdots s_m,\ \text{aligned} = t_1 \cdots t_m)\}$, conflict set C
Result: Updated rows R'

```
R' ← [ ]
for all (middle = s_1 ··· s_m, aligned = t_1 ··· t_m) ∈ R do
    for i ← 1 to m do
        if s_i ∈ C then
            u ← t_i
            t_i ← λ
            if i < m then
                t_{i+1} ← u · t_{i+1}
            else
                t_i ← u
            end if
        end if
    end for
    append (s_1 ··· s_m, t_1 ··· t_m) to R'
end for
return R'
```

The procedure terminates when no conflicts remain (i.e. $\texttt{Conflicts}(R) = \emptyset$) or when an iteration produces no change to R, in which case `SEAL` returns the current hypothesis. At that point, the learner builds a hypothesis mapping from placeholders to their observed assigned outputs using `BuildH` (Algorithm 5.4), and incorporates these learned assignments into the identity baseline via `Merge` (Definition 5.3). SEAL then outputs a standard DFT $\mathcal{T}_\Delta$ obtained by replacing each placeholder on $\mathcal{T}_\Omega$ with its learned output string under μ. Finally, the resulting output is evaluated and checked for success, which we will address in the following section.

Definition 5.3. ***Merge*** *Given H_{id} and a predicted mapping H, define $\texttt{Merge}(H_{id}, H)$ as the result of overwriting $H_{id}[s]$ with $H[s]$ whenever $s \in \texttt{keys}(H_{id})$ and the values differ.*

Algorithm 5.4: `BuildH`

Data: Rows $R = \{(\text{encoded}, \text{aligned})\}$
Result: A hypothesis mapping H (symbol $\mapsto$ set of assigned tokens)
  initialize $H[s] \leftarrow \emptyset$ for all symbols s
  for all $(\text{encoded}, \text{aligned}) \in R$ **do**
    for all (s, t) in `zip`$(\text{encoded}, \text{aligned})$ **do**
      $H[s] \leftarrow H[s] \cup \{t\}$
    end for
  end for
  return H

In what follows, we distinguish two distribution regimes for resolving placeholder assignments. The first relies solely on rightward redistribution (PushRight), and is sufficient for many feature-changing mappings. The second imposes structural constraints on edge placeholders and bounded output length, motivated by processes such as epenthesis and deletion, where unrestricted rightward shifting may fail.

5.1.3 Example

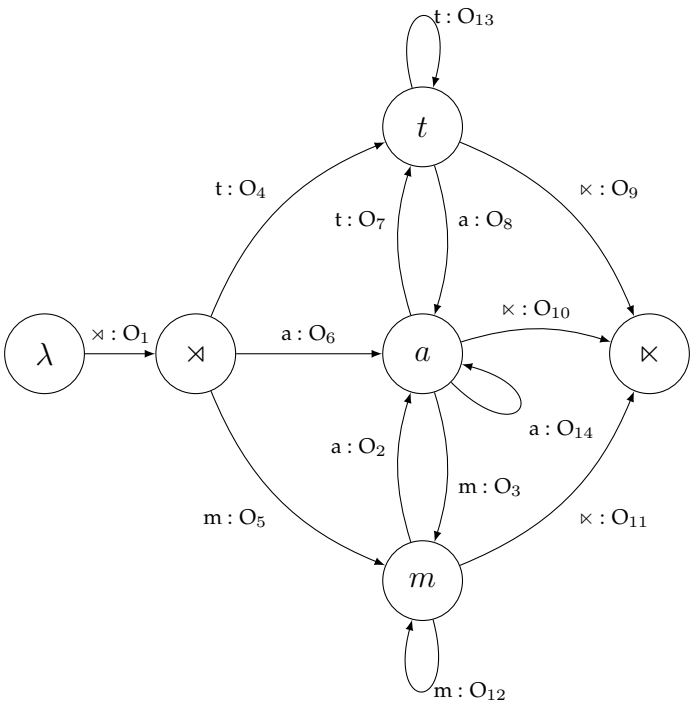


FIGURE 5.1: Subsequential transducer encoding paths.

For better comprehension and consistency, we provide an example based on a toy instance of English vowel nasalization. A transducer modeling this process is presented in Figure 5.1. Some transitions are omitted from the figure; this is strictly for readability.

Consider the following sample S:

$$\begin{array}{lcl} \text{mam} & \rightarrow & \text{mãm} \\ \text{mat} & \rightarrow & \text{mat} \\ \text{amm} & \rightarrow & \text{ãmm} \\ \text{tata} & \rightarrow & \text{tata} \\ \text{tamat} & \rightarrow & \text{tãmat} \\ \text{atama} & \rightarrow & \text{atãma} \\ \text{amata} & \rightarrow & \text{ãmata} \end{array}$$

For the transducer shown in Figure 5.1, the input mam traverses the states $\lambda{\rightarrow}\rtimes{\rightarrow}m{\rightarrow}a{\rightarrow}m{\rightarrow}\ltimes$, and the input mat traverses $\lambda{\rightarrow}\rtimes{\rightarrow}m{\rightarrow}a{\rightarrow}t{\rightarrow}\ltimes$, outputting $O_1\,O_5\,O_2\,O_3\,O_{11}$ and $O_1\,O_5\,O_2\,O_7\,O_9$, respectively. The encoded placeholder strings are then paired with their target outputs from S to form the following system of equations:

$$\begin{array}{lcl} O_1\,O_5\,O_2\,O_3\,O_{11} & = & \text{mãm} \\ O_1\,O_5\,O_2\,O_7\,O_9 & = & \text{mat} \\ O_1\,O_6\,O_3\,O_{12}\,O_{11} & = & \text{ãmm} \\ O_1\,O_4\,O_8\,O_7\,O_8\,O_{10} & = & \text{tata} \\ O_1\,O_4\,O_8\,O_3\,O_2\,O_7\,O_9 & = & \text{tãmat} \\ O_1\,O_6\,O_7\,O_8\,O_3\,O_2\,O_{10} & = & \text{atãma} \\ O_1\,O_6\,O_3\,O_2\,O_7\,O_8\,O_{10} & = & \text{ãmata} \end{array}$$

The equations are processed in the order presented. From the first equation, $O_1\,O_5\,O_2\,O_3\,O_{11} = m\tilde{a}m$, the following assignments are made. First, $O_1 = O_{\text{init}} = \lambda$ by default. Next, the one-to-one initialization in `Align1to1` (Algorithm 5.1) assigns $O_5 = m$, $O_2 = \tilde{a}$, $O_3 = m$, and $O_{11} = \lambda$.

The next equation, $O_1\,O_5\,O_2\,O_7\,O_9 = mat$, yields $O_1 = \lambda$, $O_5 = m$, $O_2 = a$, $O_7 = t$, and $O_9 = \lambda$, introducing a single conflict at O_2.

The learner continues in this manner until all equations are considered. At the end of this pass, conflicts arise at O_2, O_6, and O_8, each of which alternates between a and $\tilde{\text{a}}$. Applying `PushRight` (Algorithm 5.3) resolves these conflicts by setting $O_2 = O_6 =$

$O_8 = \lambda$ and pushing their previously assigned material to the immediately following placeholders, concatenating it with any existing output there while preserving order. After processing S, the assignments stabilize at $O_3 = \tilde{\text{a}}m$, $O_7 = at$, and $O_{10} = a$, with no remaining conflicts in this dataset. Consequently, `SEAL` (Algorithm 5.2) returns the learned transducer.

5.1.4 PRD-SEAL contra epenthesis and deletion

The `PushRight` mechanism in SEAL effectively enables the learner to withhold output material by assigning λ to conflicting placeholders, thereby permitting output-empty transitions. As will be shown in the forthcoming section, this strategy performs well for learning k-ISL functions of the feature-changing type, and it remains effective under both left-to-right and right-to-left processing directions. However, a limitation emerges when the target mapping involves *epenthesis* of an entire segment. In such cases, repeated push-right adjustments may force inserted material toward the string boundary, where it can no longer be redistributed.

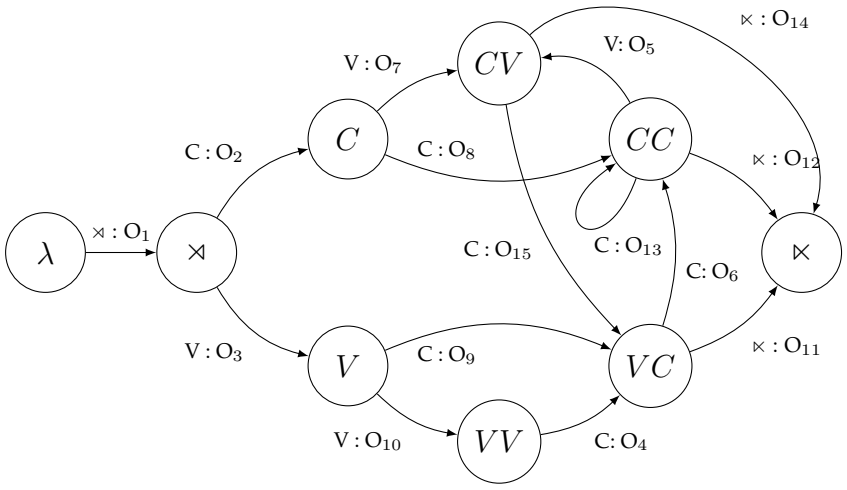


FIGURE 5.2: Path-encoding DFT over the abstract alphabet C, V, with transition outputs represented by encoded variables.

The core difficulty of this setting can be exemplified with Chukchi schwa epenthesis with R2L processing. To illustrate this, consider the transducer in Figure 5.2, representing a partial function for this process. For readability, states and alphabets are represented using C(onsonants) and V(owels), where $\Sigma = \{V, C\}$ and $\Delta = \Sigma \cup \{\text{ə}\}$. For the sample below, passing the inputs through the transducer in Figure 5.2 yields the following equations:

CCV	$\rightarrow$	CəCV	$O_1\, O_2\, O_8\, O_5\, O_{14}$	=	CəCV
VCCV	$\rightarrow$	VCCV	$O_1\, O_3\, O_9\, O_6\, O_5\, O_{14}$	=	VCCV
VCC	$\rightarrow$	VCC	$O_1\, O_3\, O_9\, O_6\, O_{12}$	=	VCC
CCCC	$\rightarrow$	CəCCC	$O_1\, O_2\, O_8\, O_{13}\, O_{13}\, O_{12}$	=	CəCCC
CVC	$\rightarrow$	CVC	$O_1\, O_2\, O_7\, O_{15}\, O_{11}$	=	CVC

Processing the first equation assigns $O_1 = \lambda; O_2 = \text{C}; O_8 = \text{ə}; O_5 = \text{C}; O_{14} = \text{V}$. After incorporating the remaining examples, the learner detects conflicts at $O_5 = \{\text{C}, \text{V}\}; O_{12} = \{\lambda, \text{C}\}; O_{14} = \{\lambda, \text{C}\}$. Following the `SEAL` procedure, these conflicting placeholders are assigned λ and their output strings are pushed to the immediately following placeholders. After one iteration, we obtain $O_5 = O_{12} = O_{14} = \lambda$. At this point, O_{14} receives C from O_5 in the first example but V from the same placeholder in the second example, resulting in a new conflict: $O_{14} = \{\text{CV}, \text{V}\}$.

In the next iteration, `PushRight` attempts to resolve O_{14} by assigning λ and pushing the displaced material rightward. However, since O_{14} is the final placeholder on that path, the displaced output has nowhere else to go and is therefore reassigned to O_{14}. A similar occurrence holds for O_{12}. As a result, the learner reaches a fixed point: the next iteration introduces no change, and `SEAL` halts even though the epenthesis pattern has not been learned correctly. Importantly, this failure is not repaired by providing a richer characteristic sample. The underlying issue is procedural, as `PushRight` resolves conflicts only by shifting output material rightward, but it does not allow an alternative repair strategy in which longer substrings are assigned earlier

so that the output can be "squeezed" into the middle portion of the path. Consequently, a new distribution method is required.

Length-Constrained Distribution (`LCD-SEAL`). Under `LCD-SEAL`, the initialization is modified so that both edge placeholders are fixed to λ by default. The learner then distributes the entire output string within the *middle* region of the encoded placeholder sequence (i.e. the encoded string with edge placeholders removed).

Since the value of k is fixed when constructing the k-ISL skeleton, we additionally impose a bound on how much output any single middle placeholder may contribute: each middle placeholder may emit a substring of length at most k. This restriction is consistent with the memory bound inherent in k-ISL DFTs. Because the transducer has access to at most k symbols of contextual information, any retained material must ultimately be realized within a bounded output contribution at some transition. Bounding substring length therefore provides a principled control over the hypothesis space while still permitting multi-symbol outputs when required.

While `LCD-SEAL` resolves the epenthesis difficulty, more complex deletion systems reveal a further tension between strictly local redistribution and boundary restrictions.

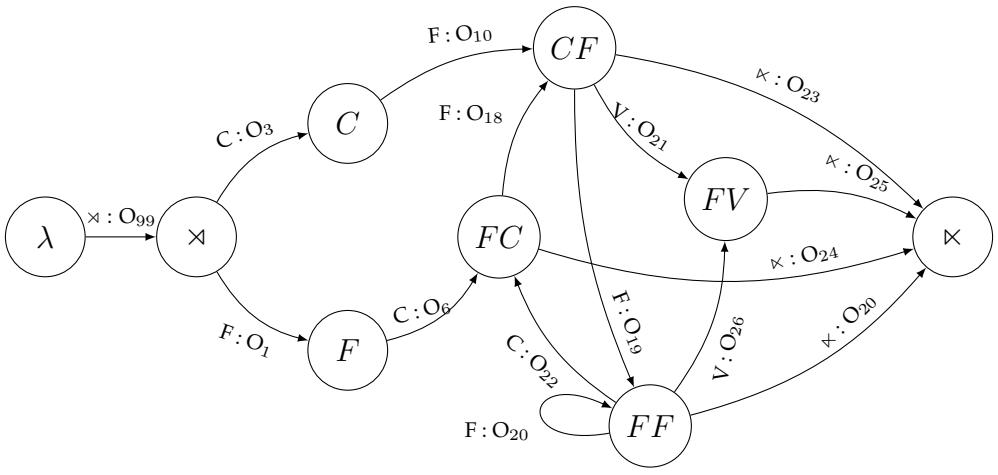


FIGURE 5.3: Path encoding DFT exhibiting two deletion processes.

Consider a toy example of two deletion processes in Samoan under R2L processing. In this language front vowels are deleted when preceded by another front vowel and

consonants are deleted word finally. In the R2L direction, the first consonant in an input string is deleted, and a front vowel is deleted when it is immediately followed by another front vowel. For presentational purposes, we restrict the input alphabet to three symbols: C (consonants), F (front vowels), and V (all remaining, non-front vowels). Thus $\Sigma = \{C, F, V\}$.

The data and equations below are generated by the transducer in Figure 5.3 and reflect these two deletion processes.

FCF	$\rightarrow$	FCF	$O_{99}\, O_1\, O_6\, O_{18}\, O_{23}$	=	FCF
CFFCFF	$\rightarrow$	FCF	$O_{99}\, O_3\, O_{10}\, O_{19}\, O_{22}\, O_{18}\, O_{19}\, O_{20}$	=	FCF
CFFCF	$\rightarrow$	FCF	$O_{99}\, O_3\, O_{10}\, O_{19}\, O_{22}\, O_{18}\, O_{23}$	=	FCF
CFF	$\rightarrow$	F	$O_{99}\, O_3\, O_{10}\, O_{19}\, O_{20}$	=	F
CFFC	$\rightarrow$	FC	$O_{99}\, O_3\, O_{10}\, O_{19}\, O_{22}\, O_{24}$	=	FC
CFFV	$\rightarrow$	FV	$O_{99}\, O_3\, O_{10}\, O_{19}\, O_{26}\, O_{25}$	=	FV
CFV	$\rightarrow$	FV	$O_{99}\, O_3\, O_{10}\, O_{21}\, O_{25}$	=	FV

With the `PRD-SEAL` mode, conflicts at O_{10} and O_{19} are repeatedly pushed rightward. This in turn creates further conflicts at O_{22} and O_{18}, and the displaced material ultimately accumulates at the right boundary. As a result, the final placeholder must reconcile incompatible assignments.

In particular, the two paths $O_{99}\, O_3\, O_{10}\, O_{19}\, O_{22}\, O_{18}\, O_{19}\, O_{20}$ and $O_{99}\, O_3\, O_{10}\, O_{19}\, O_{20}$ correspond to the outputs *FCF* and *F*, respectively. If all conflicting material is pushed to the right, then O_{20} is forced to accommodate both assignments. However, since O_{20} is final on the path, there is nowhere further to push the material, and the conflict between *F* and *FCF* cannot be resolved.

`LCD-SEAL` defined as is, with the restrictions on both right and left string boundaries, also creates problems. Consider inputs such as *FCF* and *CFFCF*, both ending in a front vowel. Under R2L processing, when a front vowel F is read, the learner should withhold its output because it cannot yet determine whether deletion occurs. A strictly boundary-constrained mode would not permit this postponed material to

be recovered, since boundary positions would be excluded from redistribution. While it is possible that for some particular sample the learner might succeed without withholding output in this position, in general this would lead to incorrect predictions once additional data are observed.

As discussed earlier, processing direction can substantially affect learning performance. In order to allow correct predictions under both L2R and R2L processing, the distribution mechanism within the `LCD-SEAL` mode must allow more flexibility. This motivates a more relaxed variant, in which output redistribution remains length-constrained, but boundary placeholders are permitted to participate when necessary.

Although this extension may appear minor, it prevents allows the possibility of outputting something other than empty string and the right word boundary, which might appear to be crucial for learning some phonological mappings. In practice, boundary participation can be enabled or disabled depending on the process under consideration, since excluding it may reduce the search space and speed up the inference when boundary interaction is not required.

In what follows, we provide a detailed description of this length-constrained distribution mechanism.

5.1.5 Length-Constrained Distribution (LCD-SEAL)

We now describe a second learning mode, which we call `LCD-SEAL` (Length-Constrained Distribution). The core idea is to impose a bounded-length constraint on placeholder outputs while controlling participation of the right boundary placeholder.

As in the original SEAL procedure, the left boundary placeholder is always fixed to λ. Under `LCD-SEAL`, the right boundary placeholder may either be excluded from redistribution (forced to λ) or allowed to participate in output distribution, depending on the process under consideration.

Let $w_o = O_{init} \cdots O_{fin}$ be an encoded placeholder string, where O_{init} represents the

first element and O_{fin} the last. Then the new learning mode starts by collecting those elements from each data point on the left side of the equation (Algorithm 5.5). Then they are forced to map to λ.

Algorithm 5.5: `Edges`
Data: Encoded sample $R = \{(w_o, y)\}$, mode $r \in \{\text{exclude}, \text{allow}\}$
Result: Right boundary placeholder set $E \subseteq \Omega$

```
E ← ∅
for all (w_o, y) ∈ R do
    if |w_o| > 0 then
        add w_o[0] to E
        if r = exclude then
            add w_o[|w_o|] to E
        end if
    end if
end for
return E
```

Definition 5.4. ***Middle*** *For an encoded placeholder string $w_o = O_1 O_2 \cdots O_m \in \Omega^*$ with $m \geq 2$,* `Middle` *is the sequence obtained by removing the first placeholder and the last one if $r =$ exclude. Formally,* $\texttt{Middle}(w_o)=[O_2, \ldots, O_{m-1}]$ *if $r =$ exclude and* $\texttt{Middle}(w_o)=[O_2, \ldots, O_m]$ *if $r =$ allow. If $m \leq 2$, then* $\texttt{Middle}(w_o) = [\,]$.

After identifying the subset of edge placeholders E, the learner constructs for each placeholder in the remaining middle portion a domain $D[O]$ of candidate output strings (see Definition 5.4). The candidate domain mapping has a bounded-length constraint: $\forall O \in \texttt{Middle},\ \forall u \in D[O], \quad |u| \leq k$. The initialization is done conservatively by intersecting candidate sets across the entire dataset, so that each placeholder retains only those substrings that are compatible with all of its occurrences.

Algorithm 5.6: `Domains`

Data: Encoded sample $R = \{(w_o, y)\}$, bound k, edge set E
Result: Domains $D : \Omega \to 2^{\Delta^{\leq k}}$

```
initialize D[O] ← Δ^{≤k} ∪ {λ} for all placeholders O
for all O ∈ E do
    D[O] ← {λ}
end for
for all (w_o = O_init ··· O_fin, y) ∈ R do
    for O_i ∈ w_o do
        if O_i ∉ E then
            restrict D[O_i] to strings v ∈ Δ^{≤k}
            such that v can be assigned to O_i in some distribution of y
        end if
    end for
end for
return D
```

Given the initial domains, the learner removes candidates that cannot participate in any globally consistent solution. Concretely, a candidate $v \in D[O]$ is eliminated if fixing $O_i = v$ makes at least one example unsatisfiable under the current domains. This pruning is iterated to convergence. If ambiguity remains, the learner performs a bounded depth-first search by selecting a placeholder with the smallest remaining domain, branching on its candidate values, and reapplying pruning after each choice.

Algorithm 5.7: `LCD-SEAL`

Data: Encoded sample R, domains D, bound k, edge set E
Result: Singleton domains D (or failure)

```
D[O_init] ← {λ}
enforce D[O] ← {λ} for all O ∈ E
D ← Prune(R, D, k, E)
if some D[O] = ∅ then
    return failure
end if
if all O ∉ E have |D[O]| = 1 then
    return D
end if
choose O ∉ E with smallest |D[O]| > 1
for all v ∈ D[O] do
    D' ← D with D'[O] ← {v}
    attempt SEAL(R, D', k, E)
    if success then
        return solution
    end if
end for
return failure
```

The LCD mode preserves SEAL's central objective of learning a single global mapping from placeholders to output strings that is consistent across the dataset. The difference is that instead of resolving conflicts through purely local push-right redistribution, `LCD-SEAL` performs bounded distribution under a k-length constraint.

The left boundary placeholder remains fixed to λ, as in the original SEAL procedure. The right boundary placeholder may either be excluded from redistribution, which is crucial for boundary-conditioned epenthesis, or allowed to participate, which is necessary for deletion systems that require temporary withholding of multiple elements and later reintroduction of output material at the final transition.

Although `LCD-SEAL` introduces additional bookkeeping compared to the baseline procedure, it continues to control the hypothesis space through the same k-based fashion bound on placeholder outputs.

5.2 Experiments

To enable a careful comparison between SEAL and SOSFIA, we evaluate SEAL on the same four phonological mappings examined in the previous chapter: English vowel nasalization, vowel shortening in Yowlumne, ə-epenthesis in Chukchi, and the opaque interaction of o-raising and final devoicing in Polish. The experimental setup is intentionally matched to the SOSFIA experiments in order to isolate differences attributable to the learner itself.

Unlike the previous chapter, we do not relearn the k value or the sufficient featural combinations Φ for each feature. Instead, we reuse the featural combinations established in earlier experiments and fix k to the value required to model the full process across features. This design choice ensures a direct comparison of sample efficiency under identical representational assumptions, allowing us to assess whether SEAL can recover the same gold transducers from synthetically generated data, and whether it can do so with fewer input–output pairs than SOSFIA.

Another reason why we exclude a fixed-k condition in these experiments is that previous results strongly suggest that adapting k to individual features is typically more *efficient* than fixing a single memory window across the board; thus, the fixed-k baseline would not provide a meaningful point of comparison.

In the chapter that follows, we evaluate SEAL in a setting where outputs and parameters are learned jointly, i.e. where the learner must infer feature-specific settings rather than being given them.

5.2.1 Setup and Evaluation

The learner is tested under two processing directions: left-to-right (L2R) and right-to-left (R2L). The overall procedure matches the previous chapter: AMßA incrementally samples input–output pairs from the generated dataset for each feature, and sampling stops as soon as the learner correctly infers the target mapping.

The appropriate evaluation criterion depends on what is the goal of learning, particularly whether the target mapping is total or partial. In these experiments, we are interested in sample efficiency under conditions where the synthetic data contain all necessary paths to cover the full transducer. We therefore evaluate SEAL by direct comparison to the gold transducer. If the learned machine mismatches the gold machine, additional datapoints are provided incrementally, and learning terminates once the learned transducer is identical to the gold transducer.

5.2.2 Results

The results for the four languages are summarized in Table 5.1, which reports the mean and standard deviation of successful sample sizes $|\mathrm{M}_F|$ across runs. Feature-specific averages $|\mathrm{M}_{(\Phi,\phi)}|$ are reported in Table 5.2, with full results provided in Appendix B.

Language	**Direction**	**Sample sizes**			
		SOSFIA		SEAL	
		Avg. $\lvert\mathrm{M}_F\rvert$	SD	Avg. $\lvert\mathrm{M}_F\rvert$	SD
English	L2R	46	3.71	34	8.64
	R2L	44	1.89	15	2.80
Yawelmani	L2R	258	62.95	312	41.15
	R2L	164	21.24	20	11.39
Chukchi	L2R	2,338	101.13	2,466	104.14
	R2L	2,412	75.47	672	78.74
Polish	L2R	8,114	145.28	8,206	61.69
	R2L	8,068	202.83	104	57.96

TABLE 5.1: Synthesized featural $|\mathrm{M_F}|$ with standard deviations (SDs) in English, Yawelmani, Chukchi and Polish for the two conditions for new learner.

As shown in Table 5.1, in the L2R condition SEAL's performance is comparable to SOSFIA's. Difference between those two learner becomes prominent in the opposite direction of processing. Both processes that target only one feature, i.e. English vowel nasalization and Yowlumne vowel shortening, were inferred from an average of 15

and 20 input–output pairs (R2L), respectively. This constitutes a substantial improvement over the best SOSFIA results for the same functions under its strongest settings (English: $|\mathbf{M}_F| = 44$; Yowlumne: $|\mathbf{M}_F| = 164$), corresponding to reductions of approximately 65% and 88%.

The largest gains occur in Polish, where SOSFIA requires over 8,000 examples to learn the opaque interaction. In contrast, SEAL successfully recovers the target mapping with an average of 104 input–output pairs in the R2L setting, and in the best case requires as few as 35 examples. These results provide direct evidence that reversing processing direction *do* yield dramatic improvements for complex functions when the learner is able to exploit the resulting reduction in non-faithful transitions. In particular, when o-raising and final devoicing are learned right-to-left, the conditioning environment becomes available before the target is encountered, substantially reducing the number of non-identical transitions that must be inferred (recall Table A.4).

Finally, SEAL succeeds on ə-epenthesis in Chukchi under the L2R condition, and LCD-SEAL succeeds under R2L processing. In the latter setting, an average characteristic sample size of $|\mathbf{M}_F| = 672$ input–output pairs was sufficient for successful inference. Although this sample size is substantially larger than those required for the feature-changing processes, the boundary-constrained mode reduces the data requirement by nearly a factor of four relative to the strongest SOSFIA results. This provides further evidence that epenthetic processes are perhaps harder to learn.

The reduction in sample sizes becomes even more apparent when we zoom in on individual features. Table 5.2 summarizes selected insufficient features for the Chukchi experiment, while reporting all insufficient features for the remaining languages. The largest gains are observed in Chukchi and Polish. For instance, inferring the feature [distributed] in Chukchi requires only 127 input–output pairs with LCD-SEAL, which is roughly seven times less than what SOSFIA needs in the corresponding setting. Similarly, for Polish, SEAL infers the [tense] mapping from an average characteristic sample on the order of dozens of examples, whereas SOSFIA

Lang.	k	Direction	Φ	$\lvert S_\Phi \rvert$	Sample sizes			
					SOSFIA		SEAL	
					Avg. $\lvert M_{(\Phi,\phi)} \rvert$	SD	Avg. $\lvert M_{(\Phi,\phi)} \rvert$	SD
ENG	2	L2R	[**nasal** cons]	39	32	4.32	27	8.90
		R2L			32	3.90	7	2.85
YAW	3	L2R	[**long**]	363	251	63.26	309	41.15
		R2L			155	21.02	17	11.39
CHUK	2	L2R	[**low**]	62	33	5.27	41	11.28
	2		[**nasal** son]	336	297	28.00	244	65.20
	2		[**distr** cor]	1,236	813	162,52	1,064	208.70
	2	R2L	[**low**]	62	37	5.5	11	2,21
	2		[**nasal** son]	336	234	49.30	50	16.31
	2		[**distr** cor]	1,236	946	71.19	127	36.68
POL	2	L2R	[**voice** son]	336	193	29.18	227	54,08
	3		[**high** voice son low]	8,250	6,984	1,262.72	7,626	629.50
	3		[**tense** voice son round]	8,250	7,176	776,65	7,716	643.00
	2	R2L	[**voice** son]	336	101	38,13	3	0
	3		[**high** voice son low]	8,250	7,156	695.43	56	41.73
	3		[**tense** voice son round]	8,250	7,240	578.20	54	29.10

TABLE 5.2: Average successful sizes for insufficient features in English (ENG), Yawelmani (YAW), Chukchi (CHUK) and Polish (POL) for the two conditions for new learner: ϕ, the predicted feature is represented in bold.

requires on the order of several thousand (over 7,000) examples in the comparable condition.

5.3 Discussion and Conclusion

This chapter introduced SEAL, a new learner for inferring k-ISL DFTs from positive data via a string-equation perspective, and evaluated it against SOSFIA on the same four synthetic benchmarks. Across English, Yowlumne, Chukchi, and Polish, SEAL achieves markedly greater sample efficiency, often by one to two orders of magnitude.

These results motivate SEAL as a strong replacement for SOSFIA in subsequent experiments, particularly in settings where available data are limited.

A central difference between the two learners concerns how they handle *onwardness*. In SOSFIA, onwardness is effectively enforced during learning by assigning transition outputs based on the longest common prefix (lcp) of observed outputs associated with a transition, thereby guaranteeing local onwardness. This makes the learner highly sensitive to output regularities, even when they reflect faithful mappings. In contrast, onwardness in SEAL is an emergent property that develops through conflict resolution: the learner begins with a conservative identity bias and only redistributes output material when the system of equations forces it to do so.

This difference is especially visible in learning over features that do not exhibit alternations. Consider the feature [consonantal] in English vowel nasalization, given the minimal sample $S_{\text{cons}} = \{(--, --), (--+, --+)\}$. From these two datapoints, one can infer that a vocalic segment remains vocalic and that a final consonant remains consonantal. Under SOSFIA, however, computing lcp over outputs associated with the transition from the left boundary to state $-$ yields $--$, because *all* outputs begin with $--$ substring. Onwardness forces the learner to assign an output string to that transition even though the mapping is faithful. SEAL, by contrast, detects no conflict in such cases and therefore preserves identity.

The experiments from this chapter also highlight an important asymmetry between feature-changing processes and epenthesis. For English and Yowlumne, which involve local alternations and feature changes, SEAL succeeds with extremely small samples under the appropriate directionality settings. Similar is true in the interaction of Polish processes, even though one of them was targeting an unnatural class (i.e. one segment). Chukchi epenthesis, by contrast, remains substantially more data-demanding. To further investigate it, in the next chapter SEAL is tested on a similar function exhibiting both epenthesis and a feature-changing process with limited data.

Finally, it is important to note that the experimental setting in this chapter assumes

total functions and logically complete synthetic paradigms, whereas natural language learners encounter sparse and uneven evidence. In realistic acquisition scenarios, learners may never observe certain logically possible strings, particularly those that violate phonotactic constraints or are otherwise unattested. This raises the broader question of whether a learner should be expected to infer outputs for unattested regions of the input space at all.

Taken together, the results in this chapter establish SEAL as a substantially more data-efficient alternative to SOSFIA, while also clarifying which aspects of the learning problem remain challenging—most notably, processes involving insertion and boundary-sensitive redistribution. In the next chapter, we therefore move beyond synthetic paradigms and test SEAL on smaller, textbook-style datasets, where sparsity and lexical gaps more closely resemble realistic learning conditions.

Chapter 6

Learning From Naturalistic Data

6.1 Introduction

In the previous chapter, we demonstrated that the proposed learner, SEAL, substantially outperforms SOSFIA in its ability to learn phonological mappings from small datasets. Across a series of experiments on synthetic data modeling phonological functions in four languages, SOSFIA required between approximately 44 and 8,000 training examples to achieve successful learning, whereas SEAL succeeded with as few as 14 to 600 examples, depending on the function and its optimal parameterization. These results establish SEAL as a markedly more data-efficient learner and provide sufficient motivation for excluding SOSFIA from further experimental consideration.

Despite these encouraging results, the experimental setup in the previous chapter exhibits several important limitations.

First, the training data were synthetically generated using alphabets of fixed and limited size. Although the segment inventories were carefully constructed to include representatives of relevant phonological natural classes, this abstraction necessarily departs from the richer segmental inventories found in natural languages. Adding segments, even that do not directly participate in the target phonological process, but that are featurally similar to the target or to the conditioning environment, may affect learnability in nontrivial ways by altering the hypothesis space available to the learner. For example, in Polish, learning alternations involving the features [high] and [tense],

as in the ɔ-raising process, requires the presence of additional vowel contrasts such as [low] or [round] to distinguish the target vowel /ɔ/ from a neutral vowel like /a/. However, if /i/ was used instead, the features [high] and [tense] alone would suffice to distinguish the target from vowels orthogonal to the transformation, effectively reducing the hypothesis space and potentially simplifying the learning problem. These considerations highlight the importance of evaluating learners on larger and more realistic segment inventories.

Second, the previous experiments assumed access to logically complete paradigms given the alphabet, creating learning conditions that are unlikely to arise in natural language acquisition. A substantial body of work has argued that children acquire linguistic generalizations from highly sparse lexical evidence, in which large portions of logically possible paradigms are unattested in the input (e.g., Berko 1958; Albright and Hayes 2003; Chan 2008; Albright 2009; Marcus et al. 1999; Gerken 2006; Yang 2006; Lignos and Yang 2016; Blevins et al. 2017). Crucially, learners nevertheless generalize productively to unseen forms, including nonce words, producing outputs that conform to the phonological grammar despite the absence of direct evidence for those forms in the lexicon.

Importantly, computational work on phonological learning likewise emphasizes that grammars can be induced from positive evidence and can support robust generalization to unattested forms even when the observed lexicon is sparse and incomplete (e.g., Goldwater and Johnson 2003; Hayes and Wilson 2008). As such, the assumption of complete paradigmatic coverage in learning experiments substantially underestimates the inferential burden faced by human learners.

Third, three of the four phonological functions examined in the previous chapter represented singleton processes learned in isolation. This again does not present very realistic learning conditions. In real language acquisition, children are not exposed to phonological processes one at a time; instead, they encounter systems in which multiple processes apply simultaneously and potentially interact. For example,

English-learning children acquire vowel nasalization in the presence of nasal place assimilation, rather than treating these processes as independent learning problems (see Vihman 1996 for discussion of early interacting patterns in child phonology). Treating phonological processes in isolation, therefore, simplifies the learning problem in ways that are not representative of real-world acquisition.

Finally, the results in the previous chapter revealed that the direction of string processing plays a crucial role in learnability. In particular, learning right-context-dependent alternations under a right-to-left processing settings allows the learner to avoid λ-transitions, that is, transitions in which output must be temporarily withheld. Notably, the only interacting processes examined in Polish both involved right-context conditioning. It remains an open question how the learner behaves when the inferred phonological grammar contains multiple processes whose conditioning environments occur on opposite sides of the target segment, a situation that is commonplace in natural languages.

This chapter addresses these limitations in two stages. First, we evaluate SEAL on smaller, more naturalistic textbook-style datasets with larger and more diverse input alphabets. These experiments require the learner to acquire multiple phonological processes simultaneously, including interacting processes with opposing contextual orientations, as well as deletion processes not examined in previous chapters. Second, where the available data permit it, we supplement the transition-level analysis with held-out evaluation. In particular, we use Serbo-Croatian to test whether a learner trained on a sparse textbook-style dataset generalizes to withheld attested forms, and we use a larger Polish CHILDES-derived dataset to evaluate whether the Polish mappings studied in Chapters 3 and 4 can be learned from frequency-ordered corpus-based input.

We refer to the first set of datasets as *naturalistic* only in a limited sense: they are more naturalistic than the synthetic paradigms used in the preceding chapters because they are adapted from phonology textbooks rather than generated from an

artificial grammar. We do not claim that they constitute natural language acquisition data, nor that they approximate natural corpora in the usual sense. They should therefore be distinguished from corpus-based acquisition data, such as CHILDES-style datasets, which contain naturally occurring interactions involving children, caregivers, researchers, and consultants. This distinction becomes important in the final empirical section of the chapter, where we move beyond textbook-style datasets and introduce a frequency-ordered Polish dataset derived from CHILDES (MacWhinney, 2000).

The main textbook-style experiments involve five languages from four language families: Kerewe and Lumasaaba (Niger–Congo; Tanzania and Uganda), Koasati (Muskogean; United States), Samoan (Austronesian; Samoa), and Serbo-Croatian (South Slavic; Serbia, Croatia, Montenegro, and Bosnia). These datasets are adapted from Ellis et al. (2022), who in turn draw on standard phonology textbooks and problem sets (e.g., Halle and Clements 1983; Roca and Johnson 1991; Odden 2014). All five datasets contain surface representations only. Since this work focuses on learning mappings from mental lexicon to surface representation, underlying forms were supplied by us following prior analyses of the respective languages.

The chapter then extends the empirical evaluation in two ways. First, Serbo-Croatian, the largest of the textbook-style datasets, is used for a held-out cross-validation study. This experiment tests whether a training-consistent hypothesis also generalizes to withheld attested forms. Second, Polish is revisited using a larger CHILDES-derived dataset ordered by frequency. This allows us to compare learning from synthetic characteristic samples with learning from a more naturalistic approximation of child-directed lexical exposure. This type of evaluation is also in line with recent work on morphological acquisition modeling, which emphasizes that models should be evaluated not only by accuracy but also by the developmental path they take as the training data increase (Kodner, 2022; Kodner et al., 2025).

The chapter is organized as follows. Section 6.2 introduces the target phonological processes and datasets, with representative examples and rules adapted from the relevant literature. Section 6.3 outlines the experimental design, and Section 6.4 describes the search strategies used to navigate the hypothesis space. Section 6.5 presents the evaluation methodology appropriate for sparse, textbook-style datasets and motivates the additional held-out evaluations. Finally, Section 6.6 reports the results, including transition-level analyses, error analyses, the Serbo-Croatian cross-validation study, and the Polish CHILDES experiment.

6.2 Processes and Datasets

This section introduces the datasets used in the experiments and the phonological processes they represent. For each language, we specify the process(es) under investigation as well as the size of the corresponding dataset.

1. nasal assimilation in Koasati (48 input–output pairs),
2. h-hardening in Kerewe (28 input–output pairs),
3. post-nasal voicing and hardening, nasal place assimilation, and velar palatalization in Lumasaaba (14 input–output pairs),
4. front-vowel and final-consonant deletion in Samoan (116 input–output pairs),
5. a-epenthesis and final l-vocalization in Serbo-Croatian (132 input–output pairs).

These five datasets constitute the main textbook-style sample for this chapter. Polish is introduced separately in the results section because it differs from these datasets in both source and function within the chapter: it is corpus-derived, frequency-ordered, and substantially larger than the textbook-style samples. We therefore treat Polish as an additional validation experiment rather than as part of the same dataset family.

The first two processes serve as controlled test cases for gradually probing SEAL's behavior in more naturalistic settings. Both nasal assimilation in Koasati and h-hardening in Kerewe involve feature-changing mappings and can be modeled as 2-ISL functions.

The Lumasaaba dataset is of particular interest because it contains multiple interacting processes exhibiting both opaque and transparent interactions. Moreover, these interactions involve both left and right contexts, making Lumasaaba a novel and challenging case study. Finally, Samoan and Serbo-Croatian represent a different class of phonological phenomena, namely deletion and insertion processes, respectively.

The datasets for all languages were adapted from Ellis et al. (2022). The original datasets contained only surface forms and glosses. For the purposes of this work, we reconstructed the underlying representations. We then collected all symbols appearing in either underlying or surface forms to define the input and output alphabets, Σ and Δ, respectively. These alphabets were subsequently used to construct the gold k-ISL deterministic finite transducers (DFTs) encoding the phonological mappings.

Across all experiments, we employ a single shared set of 25 phonological features:

$$F = \{\text{CONS}, \text{SON}, \text{CONT}, \text{DELREL}, \text{APPROX}, \text{TAP}, \text{TRILL}, \text{NASAL}, \text{VOICE}, \text{SPREADGL},$$
$$\text{CONSTRGL}, \text{LAB}, \text{ROUND}, \text{LABDENT}, \text{COR}, \text{ANT}, \text{DISTR}, \text{STRID}, \text{LAT},$$
$$\text{DOR}, \text{HIGH}, \text{LOW}, \text{FRONT}, \text{BACK}, \text{TENSE}\}.$$

The feature system follows the formalization proposed by Hayes (2009, p. 95–97). Phonemic inventories for all languages are provided in Appendix B. Each inventory includes only segments that occur both underlyingly and on the surface.

The adopted feature system allows for a maximally three-way distinction in feature values. The value 0 indicates that a segment is unspecified for a particular feature or that the feature is irrelevant for the segment in question. For example, Hayes (2009, p. 91) argues that the segment [p] is [0 back] unless it bears secondary velar articulation (e.g. [pˠ]), in which case it is [+back]. Similarly, [p] is assigned the value 0 for the

feature [strident], since within this feature theory only coronal fricatives and affricates are contrastively specified for [±].

In this work, we extend the use of the value 0 to represent underspecification. While underspecified segments could alternatively be represented using a distinct symbol (e.g. U), such a choice would expand the featural alphabet from $\{+, -, 0\}$ to $\{+, -, 0, U\}$, thereby increasing representational and computational complexity. We acknowledge that "not caring" about a feature is not formally equivalent to being underspecified for it. However, we adopt this representational simplification in order to investigate whether phonological mappings can be successfully learned from featural representations under conditions of data sparsity. A systematic comparison of different featural assumptions within the FxF framework is left for future work.

6.2.1 Koasati

Koasati is a Native American language of the Muskogean family, spoken primarily in Louisiana and Alabama (Kimball, 1991). One property of Koasati phonology is the variable surface realization of the first-person singular possessive prefix, illustrated in 10 (adapted from Odden, 2014, p. 81).

(10)

noun	**1SG.POSS**	**gloss**
tá:ta	an-tá:ta	'father'
kitilká	aŋ-kitilká	'hair bangs'
bayá:na	am-bayá:na	'stomach'
ifá	am-ifá	'dog'
t͡ʃofkoní	aɲ-t͡ʃofkoní	'bone'

As can be seen in the data above, nasal consonants assimilate in place of articulation to a following obstruent. Accordingly, the nasal portion of the possessive prefix surfaces as [n] before coronal stops, as [ŋ] before dorsal consonants, and as [ɲ] before alveo-palatal segments. In all remaining environments, the nasal surfaces as [m]. We

treat the prefix as underlyingly containing a single nasal segment that is not specified for place, notated here as N.

We formalize this as the rule in 11.

(11) Nasal Assimilation

$$N \rightarrow [\alpha \text{ place}] \; / \; \_\_ \; \left[\begin{smallmatrix} -\text{son} \\ \alpha \text{ place} \end{smallmatrix} \right]$$

Following our representational choice, we assign value 0 to the underspecified features. Concretely, N is underspecified only for those features that are required to distinguish the primary places of articulation. Coronals are described with additional features [anterior], [distributed], and [strident], while dorsals and palatal require reference to features [high], [low], [front], and [back]. The Koasati phonemic inventory is provided in Appendix C in Table C.1.

Table 6.1 summarizes the subset of features that vary across Koasati nasal consonants, together with the corresponding values assumed for the underspecified nasal N.

Feature	N	n	m	ŋ	ɲ
lab	0	-	+	-	-
cor	0	+	-	-	+
ant	0	+	0	0	-
distr	0	-	0	0	+
strid	0	-	0	0	-
dor	0	-	-	+	+
high	0	0	0	+	+
low	0	0	0	-	-
front	0	0	0	0	+
back	0	0	0	0	-

TABLE 6.1: Place features for nasal segments in Koasati.

Our underspecification criterion is straightforward: for every feature that distinguishes at least one pair of Koasati nasal consonants, N is assigned the value 0. Features that do not participate in nasal place contrasts would instead take the uniform value shared by all nasals (and are omitted here). Learning nasal assimilation under this representation therefore requires SEAL to fill in the full set of place-relevant

features in Table 6.1. In practice, this means that correct generalization may rely on combination of a few features, particularly for the ones represented in Table 6.1.

6.2.2 Kerewe

Kerewe is a Bantu language spoken in Tanzania, primarily on Ukerewe Island in Lake Victoria (e.g. Nurse and Philippson, 2003). In this language, the underlying glottal fricative /h/ surfaces as a voiceless labial stop [p] when preceded by a nasal consonant. This alternation, commonly referred to as h-hardening, is formalized in 12.

(12) h-hardening

$$/h/ \longrightarrow [p] \; / \; \left[\begin{smallmatrix} +\text{cons} \\ +\text{nasal} \end{smallmatrix} \right] \_\_$$

Examples illustrating this alternation, drawn from Odden (2014, p. 76–77), are given below in 13.

(13)

IMP	1SG.HAB	3SG.HAB	gloss
hiiga	m-piiga	a-hiiga	'hunt'
haaŋga	m-paaŋga	a-haaŋga	'create'
piinda	m-piinda	a-piinda	'be bent'
paamba	m-paamba	a-paamba	'adorn'

As the data show, /h/ surfaces faithfully when preceded by a vowel, as in the third-person habitual prefix /a-/, but is realized as [p] following a nasal consonant, as in the first-person habitual prefix /m-/. The alternation thus targets a single underlying segment rather than a broader natural class, in this respect resembling processes such as ɔ-raising in Polish discussed earlier.

Interestingly, in the feature system adopted here, Hayes (2009) characterizes [h] as [-consonantal], because it lacks an independent supralaryngeal constriction found in other consonants. The phonemic inventory of Kerewe representing the dataset used in this study is provided in Appendix C in Table C.9. Similarly to Koasati, h-hardening can be modelled with 2-ISL subsequential function.

6.2.3 Lumasaaba

Lumasaaba (also known as Masaba) is a Bantu language spoken primarily in eastern Uganda and western Kenya (Brown, 1972). The dataset for this language illustrates the interaction of several morphophonological processes. Following the analysis proposed in Brown (1972), we account for the alternations in 14 by positing four morphophonological rules, some of which feed one another: underlying velar stops are palatalized before front vowels, as seen in the diminutive form of *'frog'*; voiced fricatives undergo fortition in post-nasal contexts, as illustrated by im-beβa; an underspecified nasal segment marking noun class in indefinite forms assimilates in place to a following obstruent; and finally, obstruents are voiced after nasals.

(14)

noun-INDEF	**noun-DIM**	**gloss**
iŋ-ele	kacele	'frog'
iŋ-gafu	kakafu	'cow'
im-beβa	kaβeβa	'rat'

Three of these processes are formalized below; the nasal assimilation rule is identical to that proposed for Koasati (with the exception that the default resolution was not to labial nasal /m/ but kept underspecified. This is because the language does not allow the underspecified nasal anywhere else.). Following Brown (1972), the underspecified nasal occurs underlyingly only before another consonant, such that N + [+consonantal] is the only permissible cluster type.

(15) Voiced Fricative Hardening

$$\begin{bmatrix} +\text{cont} \\ +\text{voice} \end{bmatrix} \longrightarrow \begin{bmatrix} -\text{cont} \end{bmatrix} \; / \; \begin{bmatrix} +\text{nasal} \end{bmatrix} \_\_$$

(16) Post-nasal Voicing

$$\begin{bmatrix} -\text{son} \\ +\text{cons} \end{bmatrix} \longrightarrow \begin{bmatrix} +\text{voice} \end{bmatrix} \; / \; \begin{bmatrix} +\text{nasal} \end{bmatrix} \_\_$$

(17) Velar Palatalization

$$\begin{bmatrix} -\text{son} \\ +\text{dorsal} \end{bmatrix} \longrightarrow \begin{bmatrix} +\text{front} \\ -\text{back} \end{bmatrix} \; / \; \_\_ \; \begin{bmatrix} +\text{front} \\ -\text{cons} \end{bmatrix}$$

Several aspects of this system are worth highlighting. First, although each process can be modeled independently as a 2-ISL function, their feeding relations require the composed mapping to be characterized as a 3-ISL function. This is because the surface realization of the nasal segment depends not only on the immediately following obstruent, but also on the segment that follows that obstruent. When the obstruent is followed by a front vowel, velar palatalization applies, and the nasal surfaces as ɲ; otherwise, it surfaces as ŋ.

Second, the processes differ in the directionality of their triggering contexts. Velar palatalization is conditioned by a following segment, while post-nasal voicing and fricative hardening are triggered by a preceding nasal. As a result, underlying velar stops must, in effect, be sensitive to both leftward and rightward context in order to determine their surface realization. From a computational perspective, this creates a situation in which processing from a single direction offers no clear advantage.

Finally, the Lumasaaba dataset is the sparsest among the languages considered here but it exhibits the greatest number of interacting alternations. This makes it particularly informative for evaluating what kinds of generalizations a learner can extract from limited input.

6.2.4 Samoan

The dataset considered here represents a simplified system based on Samoan verbal inflection, rather than a full description of Samoan phonology. Samoan is an Austronesian language of the Samoan Islands (Mosel and Hovdhaugen, 1992), which are politically divided between the Independent State of Samoa and American Samoa.[1] The simplification reflects the nature of the textbook-style dataset: it isolates the forms relevant to the alternations under investigation, abstracts away from broader phonological

[1]The political division between Samoa and American Samoa reflects the colonial history of the Samoan Islands (National Park Service, 2025).

and morphophonological variation, and uses underlying forms supplied for the purposes of the present experiment. In this restricted system, verbal inflection provides evidence for two deletion processes: front vowels are deleted when preceded by another front vowel, and final consonants are deleted to comply with syllable-structure requirements (Zuraw et al., 2014), as shown in 18.

(18)

verb stem	**verb-ERG**	**gloss**
aŋa	aŋa-ia	'face'
sa:uni	sa:uni-a	'prepare'
taʔe	taʔe-a	'smash'
motu	motu-s-ia	'break'
tanu	tanu-m-ia	'cover up'

Some Samoan verb stems, when followed by a passive suffix -ia, exhibit a thematic consonant, as seen in motu – motu-s-ia in the examples above. In other verbs, no thematic consonant is present; however, when the verb stem ends in a front vowel, as in taʔe, the front vowel of the suffix is deleted. Following (Kuo, 2024), we treat the thematic consonant as a part of a reanalyzed stem. This choice should be understood as an idealized analysis adopted for the purposes of the present learning experiment, not as a claim about the correct analysis of Samoan verbal morphophonology. The goal is to define a restricted input-output mapping that allows us to test whether the learner can acquire interacting deletion processes from sparse data. Under this simplified analysis, the relevant alternations can be modeled with the following processes.

(19) Front Vowel Deletion

$$\begin{bmatrix} -\text{cons} \\ +\text{front} \end{bmatrix} \longrightarrow \text{ø} / \begin{bmatrix} -\text{cons} \\ +\text{front} \end{bmatrix} \_\_$$

(20) Final Consonant Deletion

$$\begin{bmatrix} +\text{cons} \end{bmatrix} \longrightarrow \text{ø} / \_\_\#$$

For the purposes of this study, vowel length is not treated as an independent feature. Although Samoan contrasts short and long vowels phonemically (Zuraw et al.,

2014), vowel length does not play a role in conditioning the alternations targeted in this simplified dataset. Accordingly, length distinctions are ignored in the feature representations used for learning.

6.2.5 Serbo-Croatian

Serbo-Croatian is a South Slavic language of the Indo-European family, spoken across the western Balkans, including Serbia, Croatia, Bosnia and Herzegovina, and Montenegro. Among its phonological processes, the language exhibits *a–ø* and *l–o* alternations in adjectives across genders, as illustrated in 21.

(21)

masc	fem	neut	plural	gloss
mlád	mlad-á	mlad-ó	mlad-í	'young'
óṣtar	oṣtr-á	oṣtr-ó	oṣtr-í	'sharp'
pódao	pódl-á	pódl-ó	pódl-í	'base'

In some cases, such as *mlad*, no stem alternation is observed. In others, a surface [a] appears between two consonants word-finally in masculine adjectives. This pattern has been analyzed either as underlying yer deletion (Browne, 1993, e.g.) or as vowel insertion (Simonović, 2016, e.g.). For the purposes of this study, we adopt the latter analysis. This choice is motivated by our goal of testing the model on epenthetic segments, alongside deletion processes (as in Samoan), purely for presentational reasons; we do not advocate for one analysis over the other.

In addition, word-final underlying /l/ undergoes vocalization, as in pódao. The following rules capture the processes discussed above.

(22) a-insertion

$$ø \longrightarrow \begin{bmatrix} -\text{cons} \\ +\text{low} \end{bmatrix} \; / \; \begin{bmatrix} +\text{cons} \end{bmatrix} \_\_ \begin{bmatrix} +\text{cons} \end{bmatrix}$$

(23) /l/-vocalization

$$\begin{bmatrix} +\text{cons} \\ +\text{lat} \end{bmatrix} \longrightarrow \begin{bmatrix} -\text{cons} \\ -\text{high} \\ +\text{back} \end{bmatrix} \; / \; \_\_\#$$

Both processes can be modeled using a 3-ISL function. Previous work has shown that epenthesis is among the most difficult processes to learn, as it requires the learner to modify all features. Based on our results for Chukchi, where epenthesis was learnable with approximately 600 examples, we expect learning this function to be challenging given the limited data available here.

Finally, the data in 21 also exhibit a shift of lexical stress. Since the present study focuses on the behavior of SEAL with respect to epenthesis, stress is excluded from the dataset. We return to this pattern in the next chapter, where we show how it can be learned if stress is treated as an additional feature and projected onto a separate tier.

6.3 Experimental design

Given the surface forms of the words, the first step is to generate their underlying representations. This is done manually for each language. We also manually construct k-ISL DFTs for each language that encode all alternations described in the preceding sections, following the specified rules exactly. The transducers were first validated by confirming that, given the proposed underlying representations, they correctly generate the attested surface forms.

This setup defines the full-sample condition used for the five textbook-style datasets. In the additional held-out evaluations reported later in the chapter, the same learning procedure is applied to restricted training subsets and then evaluated on withheld forms. Thus, the cross-validation experiments do not introduce a different learner or a different hypothesis space. They rather test whether the hypotheses selected under the same search procedure remain stable when the learner is trained on less than the full available sample.

For each feature $\phi \in F$, the default setup assumes that the feature is learned in isolation, i.e., only that feature is present in the input and $k = 1$. Both processing directions are available for every feature combination. As in previous experiments, both

the memory window and the number of feature combinations available to the learner can be adjusted during learning. We impose two global thresholds: a maximum number of feature combinations (set to 5), and a bound on the memory window to $k \leq 4$. While the former is data-driven, the latter is motivated by findings from cognitive science.

A recurring result across multiple domains is that humans have difficulty actively maintaining more than about four independent units at once, even when the nature of those units varies widely. Cowan (2001) reviews evidence from verbal short-term memory experiments in which participants were required to maintain sequences of spoken items (e.g., words or digits) under conditions designed to suppress rehearsal and long-term recoding, such as articulatory suppression and unpredictable recall points. Under these conditions, participants reliably recalled only three to five items, with most estimates clustering around four.

Additional evidence for small capacity limits comes from work on perception and working memory. Classic studies suggest that human perceptual and memory systems are subject to sharp limits in how many distinct items or contrasts can be represented at once (Miller, 1956; Atkinson et al., 1976; Pisoni, 1977). In visual cognition, this limit has often been argued to fall around four items. Luck and Vogel (1997) studied memory for arrays of visual objects using a change-detection paradigm: participants briefly viewed a set of colored squares or oriented bars, followed by a short delay and a test display that either matched the original or differed in one feature. Performance declined sharply once the number of objects exceeded about four. Crucially, this limit held even when each object was defined by multiple features: participants could remember the color and orientation of four objects just as well as a single feature per object, and even conjunctions of several features (color, orientation, size, gap) did not increase capacity.

Taken together, evidence from verbal tasks involving words or numbers and from visual tasks involving colors and shapes show that humans struggle to maintain more

than approximately four independent elements at once, regardless of modality or representational domain. Moreover, based on the PBase (Mielke, 2018) phonological dataset, Chandlee and Heinz (2018) show that the vast majority of attested phonological processes can be modeled within low-k ISL classes, most commonly with $k = 2$ or $k = 3$ typological evidence from phonology points to a strong preference for strictly local mappings. We therefore adopt a conservative bound of $k \leq 4$ on the learner's memory window for all experiments.

6.4 Navigating the search space

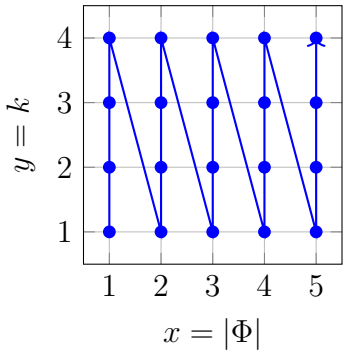
The learning procedure is parameterized by three choices: (i) the direction of mapping, (ii) the memory window size k, and (iii) the input-side featural specification Φ. In our experiments, both directions are evaluated for every (k, Φ) configuration; therefore, the structure of the search is primarily determined by how we explore k and Φ.

Both k and Φ strongly affect the size of the induced hypothesis space and, consequently, the amount of data and computation required for successful inference. Importantly, different (k, Φ) combinations can yield different functions that are nevertheless consistent with the observed training data. Among these consistent hypotheses, some may be preferable because they support better generalization and align more closely with phonological expectations, for example, by targeting a natural class rather than an accidental list of segments.

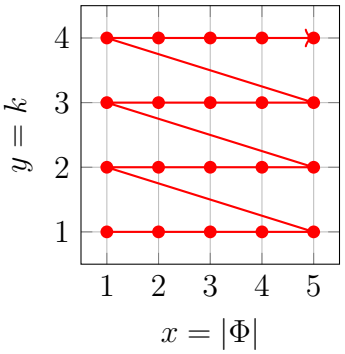
This motivates an explicit strategy for navigating the search space rather than treating parameter selection as an arbitrary choice. In particular, when several hypotheses are compatible with the data, we prefer those that are smaller and more economical unless additional complexity yields a substantial gain in explanatory power. Figure 6.1 summarizes four examples of systematic search strategies with $x = |\Phi|$ and $y = k$.

In SS1, k is increased up to a threshold while keeping $|\Phi|$ fixed. If inference fails,

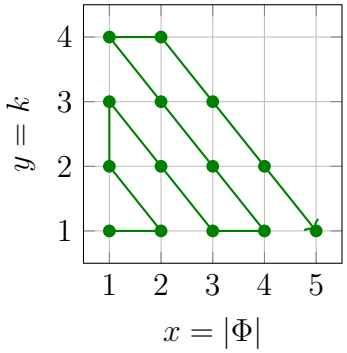
SS1: First increase k, then add feature



SS2: First add feature, then increase k



SS3: Zig-Zag Search: feature first



SS4: Zig-Zag Search: k first

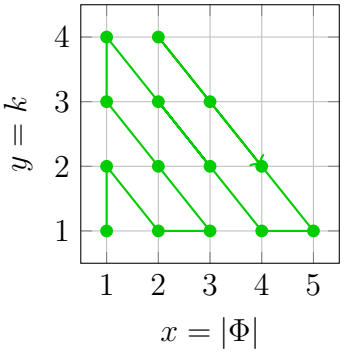


FIGURE 6.1: Four Search Strategies (SS) for increasing k (up to 4) and the number of feature combinations (up to $|\Phi| = 5$).

another feature combination is added (potentially increasing $|\Sigma|$) and then again increase k, as needed. This strategy can be computationally inefficient as increasing k tends to cause rapid growth in the state space, and can therefore become expensive early in the search.

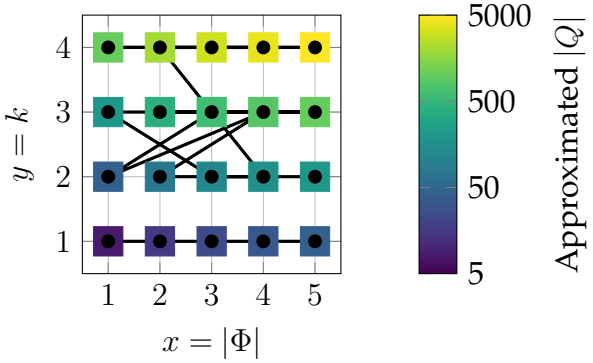
In SS2, the feature specification is expanded up to a predetermined upper bound, and only if inference fails k is increased. Compared to SS1, expanding $|\Phi|$ at fixed k is often less costly, but SS2 risks spending many iterations on k values that are too small to express the relevant context for many phonological processes.

SS3 and SS4 both increases in k and $|\Phi|$, aiming to balance between expressivity and cost. They differ only in the initial move: SS3 increases $|\Phi|$ first, whereas SS4 increases k first. In phonological applications, starting with $k = 1$ and $|\Phi| = 1$ is often extremely

restrictive and may only succeed for identity mappings or context-free dependencies. For this reason, once this setup fails, it could be beneficial for the next step to increase the memory window rather than investing heavily in richer feature alphabets while keeping $k = 1$.

While SS3 and SS4 provide a balanced and systematic way of expanding the search, they still treat all expansions as equally plausible a priori. In practice, however, the learner can use a complexity estimate to decide which configurations are worth exploring first. Concretely, we prioritize (k, Φ) settings that are expected to induce smaller machines, and only move to larger configurations when the data cannot be captured by simpler ones. This is related in spirit to MDL-based learning, where the learner trades off the complexity of the hypothesis against the extent to which it accounts for the data, preferring compact generalizations unless added structure yields a substantial improvement in fit (Rasin et al., 2021; Yang and Ellis, 2021).

SS5: Size-guided search



Unlike SS1–SS4, this strategy is not defined by a fixed geometric traversal of the $(k, |\Phi|)$ grid. Instead, the search order adapts to the predicted complexity of candidate hypotheses, with the goal of reaching a consistent and linguistically appropriate generalization without unnecessarily exploring very large machines early.

6.5 Evaluation

In earlier chapters, learner outputs were evaluated by direct comparison to the outputs of gold transducers. This strategy was appropriate in the synthetic-data setting, where the training sample was constructed to include all n-grams up to length $2k - 1$ (for a fixed alphabet and k), and thus to exercise *every* relevant path and transition in the target device.

In the more naturalistic, textbook-style datasets studied in this chapter, the same evaluation strategy is less reliable, and often infeasible. Lexicons in natural languages are sparse and systematically *incomplete*: many logically possible substrings (and thus many paths in a k-ISL transducer) are unattested even when the underlying generalization is productive. This premise is standard in experimental and computational work on phonological generalization from positive evidence (e.g., nonce-word generalization and phonotactic learning), where the learner must infer a grammar that extends beyond the observed forms (Berko, 1958; Yang, 2006; Hayes and Wilson, 2008; Zuraw, 2000; Albright and Hayes, 2003).

For this reason, we distinguish between two evaluation goals. First, we assess *sample consistency*, i.e., whether the learned transducer correctly reproduces the observed input–output pairs in the training set for each feature $\phi \in F$. Second, we treat *generalization beyond the sample* as a separate question, which we analyze manually for each learned function. In addition to identifying mismatched transitions, we also quantify how often the learned transducer (DFT_L) matches the gold transducer (DFT_G) at the transition level and report, among the matching transitions, how many correspond to *unattested* k-gram contexts in the observed data. Finally, for Serbo-Croatian and Polish, we supplement this analysis with cross-fold evaluation.

In the grammatical-inference tradition, learning from positive data can be formalized as identification in the limit from positive evidence (Gold, 1967). Under this view, a basic requirement is that a hypothesized transducer is *consistent with* the observed

positive evidence. When the goal is identification of a structured mapping from sparse positive data, evaluation on the training sample is not intended as an estimate of out-of-sample error in the sense of statistical learning theory. Rather, it serves as a check that the learned device captures the attested mapping.

Consequently, after evaluating each featural machine on its corresponding dataset, we synthesize the learned k-ISL $\text{DFT}_{(\Phi,\phi)}$ components via direct product into a single transducer over the full feature set. The resulting device is structurally equivalent, in terms of its state space, to the manually constructed segmental gold transducer. We then evaluate this recomposed machine on the original segmental sample.

Both the learned transducer and the gold transducer represent total functions over the same alphabet. The recomposed DFT_L is first tested against the full dataset S to ensure consistency at the global (segmental) level. If this condition is satisfied, we compare DFT_L and DFT_G at the level of differentiated transitions, identifying *mismatched* transitions, i.e., points at which the two devices diverge in their output assignments. Because both devices are total and the lexicon is sparse, transition-level agreement can also occur in *unattested* regions, corresponding to k-gram contexts that are not present in the observed data; we therefore track attestation of the relevant k-grams when reporting matching transitions. For each mismatched transition in the case of 2-ISL functions, we additionally test both machines on the shortest input string whose run includes that transition. Concretely, if the two machines differ on a transition (q, x, y, r), we test them on the minimal string that triggers this transition in context, i.e., $\rtimes qx \ltimes$.

For 3-ISL functions, where the number of transitions is substantially larger, we conduct an additional analysis. Specifically, we examine how the learned featural combinations Φ partition the segmental input alphabet and what implications this partitioning has for learnability. We then manually construct alternative featural combinations that partition the segmental alphabet in a manner consistent with the independently adopted phonological analysis. We test whether such finer-grained partitions

(an imposed-Φ diagnostic condition) provide any improvements.

By a partition of the segmental input alphabet, we mean the grouping into natural classes induced by a learned feature bundle. For example, suppose a successfully learned featural combination is $[\begin{smallmatrix}\text{nasal}\\ \text{cons}\end{smallmatrix}]$, with resulting abstract alphabet $\Sigma_{(\Phi,\phi)} = \{[\begin{smallmatrix}+\\ -\end{smallmatrix}], [\begin{smallmatrix}+\\ +\end{smallmatrix}], [\begin{smallmatrix}-\\ +\end{smallmatrix}]\}$, and suppose the segmental alphabet is $\Sigma = \{a, t, m, n, o, l\}$. This feature bundle partitions the segmental alphabet into three classes: vowels, nasal consonants, and oral consonants.

In the absence of independently established gold standards for phonological grammars, this structural comparison provides a principled reference point for evaluating the learner's inferred generalizations beyond the characteristic sample. Concretely, with respect to phonological mappings, this means examining how general or how restricted the learned targets and environments are in comparison to the adopted analysis. To this end, we conduct a manual linguistic analysis of the learned device to characterize the nature of the inferred generalization.

Several alternative evaluation strategies are possible. One option would be to set aside a portion of each dataset as a held-out test set. However, many of the textbook datasets are so small that further withholding examples would substantially reduce the evidence available for identifying the mapping. For example, the Lumasaaba dataset contains only 14 input–output pairs; holding out 20% of the data would leave only 3–4 examples for evaluation, yielding an estimate that is largely uninformative.

This limitation does not apply equally to all datasets considered in the chapter. Serbo-Croatian, with 132 input–output pairs, remains small, but it is large enough to support a limited cross-validation diagnostic. We therefore use it to test whether the learner can successfully predict the outputs of randomly selected held-out forms drawn from the same textbook-style dataset. The Polish CHILDES dataset provides an even stronger basis for held-out evaluation, since it contains a much larger set of frequency-ordered UR–SR pairs. In that case, cross-validation allows us to test whether the learner can recover the target mappings from corpus-derived input rather

than from the synthetic characteristic samples used in Chapters 3 and 4.

Another possible approach would be to generate additional datasets based on the observed input and output alphabets. To ensure the well-formedness of such generated samples, one could further constrain them to follow the phonotactic patterns present in the original dataset, using algorithms that operate over phonological features in a similarly bottom-up fashion, such as BUFIA (Chandlee et al., 2019). However, this strategy reintroduces the same difficulty from above: determining how restrictive the imposed function should be. As in the Selection Problem identified by Hayes and Wilson (2008), any choice of constraint strength risks either undergenerating legitimate generalizations or overgenerating unattested ones.

For these reasons, the primary evaluation for the textbook-style datasets is a qualitative, linguistically informed comparison between the learned transducer and a hand-constructed gold transducer. This approach allows us to directly assess the nature of the learner's inferred generalizations, particularly, what it allows, what it excludes, and how these choices align with independently motivated phonological analyses we chose to adopt in this work. The held-out evaluations in Serbo-Croatian and Polish provide an additional diagnostic of whether training-consistent hypotheses extend to unseen attested forms.

6.6 Results

The results are presented in three stages. First, we report the full-sample learning results for the five textbook-style datasets. All five mappings were successfully inferred in the sense that, when trained on the full available sample, the learned transducers produced the desired outputs for the observed input–output pairs. Full results for these experiments, including the learned feature combinations, the inferred k values,

and whether learning succeeded under L2R or R2L processing, are reported in Appendix C, Tables C.2–C.16.Tables C.2, C.10, C.12, C.14 and C.16. Second, we evaluate generalization beyond the full sample by comparing the recomposed learned transducer DFT_L with the gold transducer DFT_G at the transition level and by analyzing unmatched transitions. Third, we report held-out evaluation results for two datasets that allow additional quantitative testing: Serbo-Croatian and Polish. For Serbo-Croatian, we report cross-validation accuracy under both learner-selected and manually imposed feature specifications. For Polish, we report the full CHILDES-based cross-validation results, including fold-level accuracy, the identity baseline, the effect of training-set size, and the point at which SEAL begins to outperform the baseline.

6.6.1 Full-sample learning and transition-level generalization

To assess what was learned beyond the paths exercised by the training data, we conduct an additional transition-level analysis. We focus first on the transitions of the learned transducer that are identical to those of the gold transducer. We begin by estimating how much there is to learn for each language, by computing the total number of transitions in the gold transducer and counting how many of these are non-identity transitions. We then report how many transitions in the learned transducer match the gold transitions, i.e., are identical to the corresponding gold transitions (Table 6.2). Because both devices are total and the data are sparse, we additionally assess whether these matching transitions are supported by the observed lexicon: for each matched transition, we extract the relevant k-gram determined by the current state and the symbol being read, and we check whether that k-gram is attested in the dataset (Table 6.3). For example, if a matched transition is (ac, b, o, b) but no input contains the k-gram acb, then this agreement occurs in an unattested region. Finally, we generate ten out-of-vocabulary test words to probe the learned transducers' behavior *beyond*

the training sample. In four of the five languages, these tests suggest that the learned transducers DFT_L generalize beyond the observed sample, with the exception of the Lumasaaba case. We then zoom in on the unmatched transitions in DFT_L and propose what may be driving the remaining divergences.

Lang	$\lvert S\rvert$	$\lvert\delta_G\rvert$	L2R			R2L		
			$\lvert\delta_{ID}\rvert$	$\lvert\delta_{N-ID}\rvert$	N-ID %	$\lvert\delta_{ID}\rvert$	$\lvert\delta_{N-ID}\rvert$	N-ID %
Koasati	48	325	289	36	11%	306	19	6%
Kerewe	28	226	222	4	2%	197	29	9%
Samoan	116	257	42	215	84%	186	71	28%
Lumasaaba	14	2,042	1,466	576	28%	1,195	847	41%
Serbo-Croatian	132	11,662	4,033	7,629	65%	11,404	258	2%

TABLE 6.2: For each language, training-sample size (|S|) and total number of gold transitions (|δ_G|), together with the breakdown of gold transitions into identity (|δ_ID|) vs. non-identity (|δ_N-ID|) transitions under L2R and R2L processing. The non-identity count (and percentage) approximates how much of the gold mapping is non-trivial, i.e., requires learning beyond copying.

Table 6.2 summarizes how the proportion of non-identity transitions varies across languages and processing directions. In some cases, such as Kerewe in the L2R condition, the learner needs to infer only a small fraction of the gold transitions (approximately 2%), with the remainder potentially recoverable via identity mapping. In other cases, the proportion of non-identity transitions is much larger (e.g., Samoan in L2R at 84%), suggesting a substantially greater amount of non-trivial mapping to be learned.

At the same time, these percentages should be interpreted together with the size of the available sample, and the complexity of the mapping (including the number of interacting processes). For example, in the L2R condition Samoan requires learning 215 non-identity transitions for *ideal* inference from 116 examples, whereas Serbo-Croatian requires learning 7,629 non-identity transitions from 132 examples. Featural representations are expected to mitigate this difficulty in some cases. In earlier chapters we showed that featural factorization reduces the size of the relevant devices and distributes non-identity transitions across features, which in turn improves generalization from sparse data.

Finally, Lumasaaba presents the most challenging setting in this chapter, where the target mapping involves four interacting phonological processes, while the dataset contains solely 14 examples. Under these conditions, we should expect generalization to be particularly challenging.

Lang	L2R			R2L		
	% match	$\lvert\delta_{match}\rvert$	k-grams unatt	% match	$\lvert\delta_{match}\rvert$	k-grams unatt
Koasati	87%	282	223	96%	312	248
Kerewe	95%	215	172	87%	196	153
Samoan	60%	153	37	69%	175	121
Lumasaaba	14%	277	275	14%	276	274
L (imposed Φ)	68%	707	705	29%	597	595
Serbo-Croatian	4%	432	429	16%	1,822	1,819
SC (imposed Φ)	6%	726	723	36%	4,211	4,209

TABLE 6.3
Transition-level agreement between the recomposed learned transducer DFT_L and the gold transducer DFT_G under L2R and R2L processing. For each language, we report the percentage and count of gold transitions matched by DFT_L ($\lvert\delta\text{\_match}\rvert$), and, among those matches, the number of cases whose corresponding k-gram context is unattested in the observed dataset (k-grams unatt). Rows labeled "imposed Φ" report the same diagnostics under the imposed-Φ condition.

Table 6.3 reports the number of transitions in DFT_L that match those in the gold DFT_G and the corresponding percentages. For Koasati and Kerewe, we see strong performance, ranging from 87% to approximately 96% depending on the direction of processing. This is broadly what we would expect given that the number of non-identity transitions was not too high and the sample size. We also see a clear relationship between these quantities, namely, fewer non-identity transitions tend to coincide with higher match rates. For example, Koasati reaches 96% of matched transitions when only 19 transitions are non-identity in the R2L condition, whereas the rate drops to 87% in L2R when the number of non-identity transitions is nearly doubled (36).

Samoan shows a slightly different pattern. The two processing directions yield very different proportions of non-identity transitions relative to total transitions (84% in L2R vs. 28% in R2L), yet the percentage of matched transitions between the learned

and gold transducers remains relatively comparable (60% vs. 69% in L2R and R2L, respectively). One possible explanation is that the relevant generalizations may be easier to extend beyond the sample because they require fewer featural changes. Samoan case exhibits two deletion processes, one of which targets the class of consonants, which, in theory, could be described with one feature. On the other hand, nasal assimilation in Koasati required multiple features to separate the right place of articulation contexts. As we will see in a later section, the particular way in which the learned featural alphabets partition the segmental inventory may further support this kind of reasoning.

The weakest performance overall is found in Lumasaaba, which is consistent with the small sample size and the fact that multiple interacting processes are represented in the dataset. In this case, only 14% of the transitions were matched. Here we also report an additional condition. The first Lumasaaba row corresponds to the featural combination found by FxF(SEAL). We additionally ran another round of experiments in which featural combinations were selected manually, in order to test whether the type of feature combination and the number of features impact learning performance. The manual feature division was chosen so that the features separate the target sounds from their environments in a way that is intended to support the rules described at the beginning of this chapter. More details regarding this choice and its effect on learning are provided in the following section. Under these "idealized" featural combinations, accuracy increases to 68%, which highlights the importance of the chosen feature partition for learning.

Finally, in Lumasaaba as well as in Serbo-Croatian, there is another trend worth noting. In particular, nearly all of the matched transitions correspond to k-grams that do not occur in the sample (to the exclusion of empty-string transitions and a small number of singleton transitions in Serbo-Croatian). Since these matched transitions are largely unsupported by attested contexts, this suggests that identity mappings may be filling in as defaults in unattested paths. In turn, this means that match percentages for

Lumasaaba and Serbo-Croatian should be interpreted with a grain of salt. Importantly, this observations suggests that *mismatched* transitions do not necessarily imply that the learned system is uniformly incorrect in its predictions, and these statistics clearly show that more than one mapping can be compatible with the available evidence.

Below we provide ten manually constructed input strings for each language and report how the learned transducers behave on these test items. The Lumasaaba results are not included because the learned transducer for that function failed on essentially every example, with the exception of a small set of carefully chosen items that differed from an attested form by exactly one segment and only under a very specific configuration. In particular, replacing a front mid vowel with the front high vowel [i] yielded the expected result only when the vowel was preceded by a velar consonant, which itself was preceded by a nasal. In other words, the learned Lumasaaba system fits the training data but does not appear to support reliable predictions outside of the sample. Given that there were only 14 examples to learn from, this outcome is not too surprising. Moreover, as the phonemic inventory in Table C.11 illustrates, there was limited scope for generalization. Concretely, the inventory contains a representative of one consonant for some major places of articulation, which restricts how much the learner can exploit featural decomposition. This lack of variety matters because nasal assimilation requires place information. A similar issue arises for palatalization, where the triggering context is also extremely limited (in practice, only /k appears in the relevant environment).

For the remaining four cases, the new input strings were mapped to the expected outputs, following the rules described for the respective languages. In Koasati, the R2L condition yields perfect predictions, while L2R diverges on a single example in which the underspecified nasal occurs in word-final position. We are hesitant to interpret this as a failure per se, because in Koasati the underspecified nasal N is expected in prefixes and does not normally occur word-finally. This, in turn, raises a broader question about how "gold" the gold transducer is for each language, and about whether

representing phonological mappings as total functions is always the best choice for modeling generalization. We return to these issues in the discussion section of this chapter.

K O A S A T I					
L2R			R2L		
aNkowa	↦	aŋkowa	awokNa	↦	awokŋa
kaNiwa	↦	kamiwa	awiNak	↦	awimak
haNtølakf	↦	hantølakf	fkaløtNah	↦	fkaløtnah
lanNbø	↦	lambø	øbNnal	↦	øbmal
føaaNtø	↦	føaantø	øtNaaøf	↦	øtnaaøf
wiaNpy	↦	wiampy	ypNaiw	↦	ypmaiw
øfkyøN	↦	✗	NøykfØ	↦	NøykfØ
baNtokal	↦	bantokal	lokatNab	↦	lokatnab
htølakf	↦	htølakf	fkaløth	↦	fkaløth
nlohaNø	↦	nlohamø	øNaholn	↦	ømaholn

A similar issue was observed in Kerewe. Here the target of the change was /h/, and the dataset contained no examples in which /h/ occurred word-finally. In the L2R condition this was not problematic, because the default identity behavior effectively filled in the unattested case. In the reversed processing condition, however, the default identity behavior is precisely what caused the error, because default-identity filling operates at the transition level rather than at the path level. This distinction can create processing effects in which information about particular features is either lost or doubly output. We return to this issue in more detail later.

KEREWE					
L2R			R2L		
diŋki	↦	diŋki	ikŋid	↦	ikŋid
mhegmuem	↦	mpegmuem	meumgehm	↦	meumgepm
eŋmhiem	↦	eŋmpiem	meihmŋe	↦	meipmŋe
ŋpkidbe	↦	ŋpkidbe	ebdikpŋ	↦	ebdikpŋ
edimhad	↦	edimpad	dapmide	↦	dapmide
mpihube	↦	mpihube	ebuhipm	↦	ebuhipm
dgie	↦	dgie	eigd	↦	eigd
eŋkimhe	↦	eŋkimpe	ehmikŋe	↦	epmikŋe
uidmih	↦	uidmih	himdiu	↦	✗
hikki	↦	hikki	ikkih	↦	ikkih

Samoan was the only case in which we achieved 100% correct predictions on out-of-sample words. This is particularly striking given that the mapping involves two deletion processes affecting every feature. The processes do not interact with each other, however, which likely contributed to this outcome.

SAMOAN					
L2R			R2L		
moʔamuv	↦	moʔamu	vumaʔom	↦	umaʔom
taŋon	↦	taŋo	noŋat	↦	oŋat
toovuŋ	↦	toovu	ŋuvoot	↦	uvoot
taomiva	↦	taomiva	avimoat	↦	avimoat
leleiv	↦	lele	vielel	↦	elel
saunimev	↦	saunime	veminuas	↦	eminuas
paʔip	↦	paʔi	piʔap	↦	iʔap
unniaev	↦	unniae	veainnu	↦	veainnu
puleip	↦	pule	pielup	↦	elup
tuvul	↦	tuvu	luvut	↦	uvut

Finally, Serbo-Croatian could not correctly predict one of the test words in the R2L condition, regardless of whether the feature set Φ was learned by FxF(SEAL) or manually constructed. In the L2R condition, the manually curated feature division improved performance on two cases, which again suggests that better feature combinations can lead to better generalization. At the same time, this supports the broader point that there can be multiple solutions compatible with the available evidence, and that some are clearly preferable to others. This observation motivates the need to think more carefully about how to support the learner in selecting feature combinations that are most efficient to generalization, which is a non-trivial problem.

SERBO-CROATIAN							
L2R				R2L			
		L_Φ	I_Φ			L_Φ	I_Φ
bidel	$\mapsto$	bideo	✓	ledib	$\mapsto$	oedib	✓
vebal	$\mapsto$	vebao	✓	labev	$\mapsto$	oabev	✓
rasn	$\mapsto$	rasan	✓	nsar	$\mapsto$	nasar	✓
krubok	$\mapsto$	krubok	✓	koburk	$\mapsto$	koburk	✓
krubl	$\mapsto$	✗	krubao	lburk	$\mapsto$	✗	✗
benk	$\mapsto$	benak	✓	kneb	$\mapsto$	kaneb	✓
titn	$\mapsto$	titan	✓	ntit	$\mapsto$	natit	✓
bes	$\mapsto$	✗	bes	seb	$\mapsto$	seb	✓
rabel	$\mapsto$	rabeo	✓	lebar	$\mapsto$	oebar	✓
krabr	$\mapsto$	krabar	✓	rbark	$\mapsto$	rabark	✓

In what follows, we zoom in on what went wrong by examining the mismatched transitions in detail. We characterize the errors they induce and propose concrete directions for improvement, including changes to the representation and to the learning setup that may better support generalization from sparse data.

6.6.2 Error analysis of the unmatched transitions

Koasati

As expected, the only features that were *insufficient* (i.e., required $k > 1$ and additional information from other features) were precisely the place-related features listed in Table 6.1. In every such case, SEAL needed one or two additional features to gather informative enough input information. The resulting sufficient combinations are summarized in Table 6.4.

Feature ϕ	Sufficient set Φ	k
lab	[lab, tap]	2
cor	[cor, tap]	2
ant	[ant, nasal]	2
distr	[distr, nasal]	2
strid	[strid, nasal]	2
dor	[dor, ant, tap]	2
high	[high, delrel, nasal]	2
front	[front, delrel, nasal]	2
back	[back, ant, son]	2

TABLE 6.4: Results for learning the insufficient features in Koasati nasal assimilation

The dominant sources of error in Koasati, as in the remaining languages, fall into one of three types: (i) λ-transitions (i.e., postponing output until more relevant context is within the memory window), (ii) sparse coverage of relevant contexts in the training data, and (iii) insufficiently general identification of the target, the triggering environment, or both.

To illustrate error type (iii), consider the partition of the input alphabet induced by the feature ϕ = [lab] with the learned combination $\Phi = \left[\begin{smallmatrix}\text{lab}\\ \text{tap}\end{smallmatrix}\right]$. This combination separates labial consonants from the remainder of the inventory and, crucially, singles out the underlying underspecified nasal N as the intended *target*. For most insufficient features, the partition of the original segmental alphabet induced by Φ correctly

isolates the target, rather than grouping it with other nasal consonants as a broader natural class.

The main exceptions are the features [high] and [front]. In these cases, the learned combinations Φ expand the target class to include additional nasals. As a result, in the L2R setting, DFT_L withholds output in contexts where the gold device would not, reflecting an overgeneralization of the target environment.

Two additional, feature-specific overextensions illustrate the same general point. The feature [dor] does not distinguish dorsal obstruents from the approximant /w/ as the learned feature set Φ contains element int he resulted alphabet, particularly $\left[\begin{smallmatrix}+\text{dor}\\ 0\,\text{ant}\\ -\text{tap}\end{smallmatrix}\right]$, that corresponds to both {k, w}. This lack of distinction causes incorrect generalization evident in DFT_L where N will be assimilating to /w/ and producing + value for that feature, whereas the DFT_G assigns [- dor] to the underspecified nasal because it predicts the default labial nasal in such environments.

Likewise, [ant] expands the coronal trigger class to include l, since coronal obstruents and l share $\left[\begin{smallmatrix}+\text{ant}\\ -\text{nasal}\end{smallmatrix}\right]$ under the learned partition.

For further analysis, we processed the learned DFT_L on some arbitrary strings that contain the path represented in the transitions that differ from the corresponding ones in DFT_G . The failure of isolating underspecified nasal in the input, shows up immediately when testing on such strings. For example, for an input an, DFT_G outputs an, but DFT_L fails to realize the withheld outputs for features [front] and [high] (due to the broader learned target definition). However, because the sample contains no instances of a nasal in word-final position, upon reading the right word boundary DFT_L outputs empty string, which is recognized as the identity mapping for transitions to the final state $\ltimes$.

A related boundary effect arises in the transitions coming out of the λ state. The learner was never exposed to strings beginning with a nasal in R2L processing. Consequently, when tested on inputs such as nb, the learned machine again defaults to identity when processing the first element of the input. Then when in the state n,

features [front] and [high] have learned that we need to return the withheld values together with the values of the element being processes. That leads to *extra* feature values material for [front] and [high] resulting in an output that does not correspond to any full segment. That would be an example of (i) type error.

Finally, sparsity (error type (ii)) also affects the *default resolution* of underspecification. In the gold analysis, N surfaces as [m] in the elsewhere case, but in the learning process there is often no pressure to posit this default when the data do not force it. For example, on an input such as Nl, DFT$_G$ maps the underspecified nasal to [m], whereas DFT$_L$ leaves it unchanged.

Importantly, the composed DFT$_L$ still reproduces the original segmental sample. We then compare transition-by-transition outputs, we find substantially more mismatches under left-to-right processing than under right-to-left processing (Table 6.5).

	Mismatched $\lvert\delta\rvert$	
$\lvert\delta_G\rvert$	**L2R**	**R2L**
325	43	13

TABLE 6.5: Koasati: number of incorrect transitions relative to the gold 2-ISL DFT.

Under the R2L processing, the mismatch count drops substantially (Table 6.5). The main reason behind it is that the triggering obstruent is encountered *before* the target nasal, so the machine can often emit the relevant place features without resorting to waiting transitions. As a result, there are fewer transitions whose outputs must be inferred indirectly, and fewer places where the identity default can cause issues.

The R2L errors are largely of the same type as in L2R, but compressed into fewer problematic transitions. For example, because [high] still fails to distinguish underspecified N from fully specified nasals, encountering a dorsal obstruent k incorrectly forces [+high] on *any* following nasal. Similarly, for an input such as nN, DFT$_G$ yields nm (with [m] being the elsewhere case), whereas the learned system outputs + value

for [coronal] because the feature's alphabet partition does not separate coronal obstruents from coronal sonorants.

The only insufficient feature that was perfectly mapped is [back]. It is not surprising as the environment for changing N to palatal ɲ is accurately excluded and the underlying value [0 back] of the underspecified N does not change in any other contexts – hence it can be solved with assuming default identity.

Overall, these results underscore how much SEAL can recover from very spare data. Koasati nasal assimilation is learned from only 22 examples, yet the learner still converges when tested on samples for individual features and provides largely correct behavior.

Detailed transducer statistics and the induced inventory partitions for the insufficient-feature combinations are reported in Appendix C (Tables C.6–C.8).

Kerewe

Learning the hardening process in Kerewe resulted in five features being classified as insufficient (Table 6.6). All but one required information from one additional feature in order to converge on the sample. The exception was [spread glottis], which was sufficient on its own provided that the memory window was extended to $k = 2$. This is not unexpected, given that the feature uniquely identifies the target segment h as the only segment specified [+spread glottis] in the inventory, and thus can isolate it without assistance of any other features. However, [spread glottis] (like the remaining insufficient ones) does not independently distinguish the relevant environment for the change, at least not in the expected way under the current analysis of this phonological process. Consequently, errors are expected to arise once the per-feature transducers are composed.

When evaluated on their respective featural samples, all learned feature-level transducers produce the correct outputs. Moreover, their synthesis correctly reproduces the

Feature ϕ	Sufficient set Φ	k
cons	[cons, approx]	2
cont	[cont, approx]	2
delrel	[delrel, approx]	2
spreadgl	[spreadgl]	2
lab	[lab, cons]	2

TABLE 6.6: Results for learning the insufficient features in Kerewe /h/-hardening

segmental training sample. As observed in Koasati, different directions of processing amount to different proportions of *unmatched* transitions (Table 6.7).

	Mismached $\lvert\delta\rvert$	
$\lvert\delta_G\rvert$	**L2R**	**R2L**
226	30	11

TABLE 6.7: Kerewe: number of incorrect transitions relative to the gold 2-ISL DFT.

In contrast to Koasati, the hardening trigger in Kerewe is located to the *left* of the target. Accordingly, fewer errors arise under L2R processing, where the relevant context is encountered before the target. By contrast, the R2L processing requires postponing output decisions until sufficient left context becomes available, leading to a greater number of mismatched transitions.

Some discrepancies are the result of how the learned featural combination partitions the segmental alphabet. Let us consider the input alphabet $\Sigma_{\left(\left[\begin{smallmatrix}\text{delrel}\\ \text{approx}\end{smallmatrix}\right],\text{delrel}\right)}$ in Table 6.8

Unlike [spread glottis], the learned featural combination for [delayed release] does not differentiate between segments such as $\widehat{\text{tʃ}}$ and h. As result, the learned function is expected to make wrong predictions by generalizing the target of the change to also include palato-alveolar affricate. To make it concrete, when both transducers tested on an input /$\widehat{\text{ntʃ}}$/, DFT$_G$ outputs identity string [$\widehat{\text{ntʃ}}$] while DFT$_L$ changes the [+ delrel]

$\Sigma_{\left(\left[\begin{smallmatrix}\text{delrel}\\ \text{approx}\end{smallmatrix}\right],\text{delrel}\right)}$	Matched $\sigma \in \Sigma_{seg}$
$\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$	{a, e, i, u}
$\left[\begin{smallmatrix}-\\-\end{smallmatrix}\right]$	{b, d, g, k, p}
$\left[\begin{smallmatrix}+\\-\end{smallmatrix}\right]$	{$\widehat{\text{tʃ}}$, h}
$\left[\begin{smallmatrix}0\\-\end{smallmatrix}\right]$	{m, n, ŋ}

TABLE 6.8: Featural alphabet learned for [delayed release] (excluding boundary markers $\ltimes$ and $\rtimes$) with their corresponding segmental elements.

value to [-delrel] value for the second element, in the same way it would for /nh/, creating an element that is outside of Δ_{seg}.

Similar systematic errors types emerge for other features. For example, [labial] in combination with [cons] isolates the class of labial segments rather than just the nasal stop /m/, thereby expanding the triggering context. As a result, the synthesized DFT_L predicts that /h/ will become [+ labial] even when it follows oral labial stops, including /p/.

Similarly, the feature sets Φ learned for features [cons], [cont], and [spread glottis] distinguish the target /h/ but only partition the remainder of the inventory into a coarse vowel–consonant contrast. For these features, any consonant following /h/, in R2L setting, may be interpreted as a valid triggering environment. On an input such as hd, the gold device yields identity, whereas the learned device incorrectly emits feature values for [cons], [cont], and [spreadgl] matching their correspondents in /p/.

Because the training data does not contain any word-initial /h/, transition from λ to h state also defaults to identity in DFT_L in R2L direction. When combined with later non-identity outputs, this result in duplicated material for those features. For example, when processing /hd/, for feature [cons], DFT_L predicts, in that order, [- cons] for the first element, which is the glottal fricative (identity assumption due to lack of evidence in the sample), then when processing /d/ DFT_L outputs the *supposedly* withheld output and returns ++ as /d/ is included as the context for hardening. Errors of such type arise not from misidentifying the target, but from learning an overly broad

contextual partition.

Across both processing directions, the same enriched feature sets Φ were inferred. The difference lies primarily in how often output must be postponed. Under L2R processing, the left context is immediately available, eliminating most λ-transitions and preventing length mismatches in feature outputs. Under R2L processing, by contrast, postponed outputs are more frequent, and the interaction with default identity behavior leads to additional discrepancies. In this respect, the Kerewe results mirror the general pattern observed for Koasati: when the direction of processing aligns with the direction of the conditioning environment, the learned system more closely approximates the gold transducer.

Samoan

In Samoan, all features required more information outside of its own except [front]. This feature alone partitions the segmental alphabet into consonants, front vowels, and back vowels, which is precisely the distinctions required for the two deletion processes under consideration. As a result, $\text{DFT}_{L\ ([\text{front}], front)}$ is the smallest, with just 6 states and 17 transitions to learn.

The overall number of mismatched transitions is substantially higher than in Koasati and Kerewe (Table 6.9). This is expected for two independent reasons. First, the grammar involves two deletion processes that do not interact, which can increase the number of contexts and targets the learner must account for. Second, deletion, similarly to insertion, affects the realization of *all* features at once, which increases the number of non-identity transitions overall.

	Mismached $\lvert\delta\rvert$	
$\lvert\delta_G\rvert$	**L2R**	**R2L**
257	104	82

TABLE 6.9: Samoan: number of incorrect transitions relative to the gold 2-ISL DFT.

Consonant and front vowel deletions in Samoan are conditioned by right and left environment which does not make it immediately obvious to predict which direction of processing might be more efficient. Table 6.9 shows that learning in the R2L direction causes less erroneous transitions with respect to the DFT_G . That suggests that boundary environments allow more precise inference, which is not unreasonable as both $\rtimes$ and $\ltimes$ are not composed of any featural content.

A systematic error arises under L2R processing in transitions involving two consecutive consonants. The training sample contains only substrings of type CV and VV so no inputs exhibit adjacent consonants. Consequently, when the learner encounters a transition from a consonant state while reading another consonant, it defaults to identity, as imposed by the bias in the absence of evidence. In practice, this results in one consonant being effectively lost. For example, on an input /fp/, the first consonant may be withheld pending contextual confirmation, but since no CC sequences are attested, the withheld information is never properly resolved. The output, therefore, surfaces as just [p], reflecting loss of the preceding consonant's features specification.

Feature [HIGH] introduces a different type of difficulty. Under the learned partition, the velar nasal ŋ is isolated from the remaining consonants, and the high front vowel /i/ is not grouped together with the mid front vowel /e/, both of which trigger deletion. As a result, it is harder for SEAL to generalize. For instance, if the data contains no path from ŋ to e, then on input ŋe the withheld value for [high] for the first input element is never returned in DFT_L , yielding an incomplete featural specification. Similarly, sequences of identical vowels, such as ee are not properly processed by the learned machine. The first vowel is emitted as expected, while for feature [high] reading another underlying [e] does not trigger its deletion because the sample did not contain ee substrings, as opposed to for example, ei for which DFT_L correctly inferred deletion of the second consonant. This particular example shows how too fine partition of segmental alphabet may have undesirable effects. As a result, two values for feature [high] are returned while every other feature properly identified the deletion.

By contrast, the remaining insufficient features correctly group the front vowels /e/ and /i/ together and therefore generalize more successfully with respect to the identification and generalization of the environment. The limitations of [high] appear to stem from the representational system itself: within the current feature subset, the velar nasal is the only consonant specified [+high], making it impossible to group it naturally with other consonants. A different feature system might permit a more accurate generalization, but under the present representation such grouping is unavailable.

R2L processing exhibits a related but structurally distinct pattern. In several cases, the learned and gold transducers differ in how outputs are distributed across transitions, even when the overall string-level output matches on the training sample. This reflects the output distribution mechanism in LCD-SEAL mode. Multiple output distributions are available from the very beginning, the learner may converge on different (from gold one) transducer that is nonetheless functionally equivalent to the target mapping.

Consider the input ue. In DFT_L , the high vowel withholds its output for [high], not for phonologically motivated reasons, but simply because this distribution constitutes a viable hypothesis under the learner's search space. Upon reading e, the feature value [-high] is emitted, which corresponds to identity for that element, as no such paths were observed for this context. Finally, upon reading $\ltimes$, the learned transducer emits another [-high], this time consistent with the earlier withholding of the output associated with e. Even though this distinct but majorly equivalent mapping may work for inputs within the sample such as e?e, we can expect some inconsistencies when tested on previously unseen data points. In particular, the combination of unmotivated output withholding and the learner's default identity behavior on unattested paths yields the surface form oe for the input /ue/, rather than the expected identity mapping.

Despite the relatively high number of transition mismatches, only a small subset leads to genuinely incorrect predictions on complete strings. Of the 82 R2L mismatches, only four transitions produce undesired outputs when tested on short strings

covering path represented in the mismatched transition. Those are:
(e, e, X, e), (e, i, X, i), (e, N, X, N), (u, e, X, e). Given that the process involves deletion of every single feature value, this degree of convergence, following this interpretation of results, is unexpectedly high.

Lumasaaba and Serbo-Croatian

The remaining two functions are represented by substantially larger transducers. The Lumasaaba processes are modeled by a 3-ISL function with 2,042 transitions, while the Serbo-Croatian transducer contains 11,662 transitions. Consequently, a detailed qualitative analysis of individual transitions, as conducted for Koasati, Kerewe, and Samoan, would be much more difficult to conduct and to present comprehensively.

Instead, we focus on a trend that emerged in the earlier analyses in which the learned featural alphabet partitions the segmental input vocabulary (Σ_{seg}) has a significant impact on generalization quality. Incorrect identification of the target or environment, either because those classes are too broad or too general, leads to wrong predictions. Here we therefore examine closer the relationship between Σ-partitioning and the number of mismatched transitions relative to the gold machine.

We compare the learned featural combinations Φ against manually designed combinations intended to reflect the independently adopted analysis of each process. The goal is to determine whether a more linguistically appropriate partition of Σ_{seg} improves performance. For each function, we report the *learned* and *imposed* combinations, the resulting size of $\Sigma_{(\Phi,\phi)}$, and the number of mismatched transitions. A summary is provided below while full results appear in Appendix C.

In both languages, the number of mismatched transitions is relatively high. In Lumasaaba (L2R), over 86% of transitions differed from those in DFT_G . When manually curated feature combinations are imposed, the number of mismatches decreases substantially, particularly in the R2L setting, reducing the error rate by nearly 20%.

	Total $\lvert\delta_G\rvert$	Learned Φ		Imposed Φ	
		L2R	**R2L**	**L2R**	**R2L**
Lumasaaba	2,042	1,765	1,766	1,335	1,445
Serbo-Croatian	11,662	11,230	9,840	10,936	7,451

TABLE 6.10: Number of wrongly predicted transitions for Lumasaaba and Serbo-Croatian transducers with learned and imposed Φ

A similar pattern holds for Serbo-Croatian. R2L processing yields better performance overall, and imposing linguistically motivated feature combinations reduces errors from 8,084 to 5,983 transitions (approximately 70% to 51%), again an improvement of about 20%.

Although these error rates may appear high, it is important to consider the structural complexity of the functions involved. Both languages require 3-ISL DFTs, unlike the earlier 2-ISL cases. This substantially increases the number of transitions to be learned. Lumasaaba involves four distinct phonological processes, yet the dataset contains only 14 input–output pairs. Serbo-Croatian includes epenthesis, which was already shown in Chapter 5 to require substantially more data for reliable identification. In Chukchi, learning epenthesis alone required over 600 examples on average. Here, Serbo-Croatian combines epenthesis with an additional process of l-vocalization, but is trained on only 132 examples. Under such data constraints, substantially higher error rates are expected.

To better understand the source of those mismatched transitions, we now examine how the learned and imposed feature sets Φ partition the segmental alphabet. As an illustration of Σ-based generalization, consider feature [voice] in Lumasaaba.

In the learned Φ for [voice], the underspecified nasal N (the trigger of obstruent voicing) is grouped together with the labial fricative β and the oral sonorant l. This partition fails to isolate the intended environment class and therefore predicts overgeneralization. Under such a natural class division, voiceless obstruents are predicted to voice before all voiced consonants, rather than only before nasals. The imposed

Learned Φ		Imposed Φ	
$\Sigma_{\left(\left[\begin{smallmatrix}\text{voice}\\\text{cons}\end{smallmatrix}\right],\text{voice}\right)}$	Matched $\sigma \in \Sigma_{seg}$	$\Sigma_{\left(\left[\begin{smallmatrix}\text{voice}\\\text{cont}\\\text{nasal}\end{smallmatrix}\right],\text{voice}\right)}$	Matched $\sigma \in \Sigma_{seg}$
$\left[\begin{smallmatrix}+\\+\end{smallmatrix}\right]$	{β, l, N}	$\left[\begin{smallmatrix}+\\+\\-\end{smallmatrix}\right]$	{β, l, w, i, u, e, o, a}
$\left[\begin{smallmatrix}-\\+\end{smallmatrix}\right]$	{f, t, k}	$\left[\begin{smallmatrix}-\\+\\-\end{smallmatrix}\right]$	{f}
$\left[\begin{smallmatrix}+\\-\end{smallmatrix}\right]$	{w, i, u, e, o, a}	$\left[\begin{smallmatrix}-\\-\\-\end{smallmatrix}\right]$	{t, k}
		$\left[\begin{smallmatrix}+\\-\\+\end{smallmatrix}\right]$	{N}

TABLE 6.11: Featural alphabet learned for [voice] excluding boundary markers ⋉ and ⋊ for Lumasaaba.

combinations increased the featural alphabet by only one, suggesting that identifying more accurate classes is not that much more expensive. Examples like this show that terminating the learning procedure at the first consistent solution may be premature. Incorporating biases that help evaluate which learned class is phonologically more adequate, together with phonotactic constraints, could guide the learner toward more robust and linguistically appropriate partitions.

Learned Φ		Imposed Φ	
$\Sigma_{\left(\left[\begin{smallmatrix}\text{approx}\\\text{trill}\end{smallmatrix}\right],\text{approx}\right)}$	Matched $\sigma \in \Sigma_{seg}$	$\Sigma_{\left(\left[\begin{smallmatrix}\text{approx}\\\text{lat}\end{smallmatrix}\right],\text{approx}\right)}$	Matched $\sigma \in \Sigma_{seg}$
$\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$	{a, e, i, o, u}	$\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$	{a, e, i, o, u}
$\left[\begin{smallmatrix}-\\-\end{smallmatrix}\right]$	{b, d, g, k, p, s, t, t͡ʃ, v, z, ʂ, ʐ, m, n}	$\left[\begin{smallmatrix}-\\-\end{smallmatrix}\right]$	{b, d, g, k, p, s, t, t͡ʃ, v, z, ʂ, ʐ, m, n}
$\left[\begin{smallmatrix}+\\-\end{smallmatrix}\right]$	{l, j}	$\left[\begin{smallmatrix}+\\+\end{smallmatrix}\right]$	{l}
$\left[\begin{smallmatrix}+\\+\end{smallmatrix}\right]$	{r}	$\left[\begin{smallmatrix}+\\-\end{smallmatrix}\right]$	{r, j}

TABLE 6.12: Featural alphabet learned for [approximant] (approx) excluding boundary markers ⋉ and ⋊ for Serbo-Croatian.

A parallel issue arises in Serbo-Croatian for features such as [approximant] and [trill]. In both cases, the learned partition fails to distinguish between j and l. Table 6.12 illustrates this for [approximant].

Because j is not excluded from the target class, the learned machine predicts that words ending in j would undergo the same alternation as l, receiving [0 approx] and [0

trill] word-finally, rather than retaining their underlying values [- approx] and [- trill]. This constitutes an overgeneralization of the target.

Interestingly, the [approximant] transducers in both the learned and imposed conditions are identical in size: the featural alphabet contains six elements, and the resulting state space consists of 23 states. This example demonstrates that it is possible to improve generalization without increasing the structural complexity of the learned machine in any way.

These cases reinforce the broader observation that the partition of Σ induced by Φ directly affects learnability. Each feature implicitly partitions the alphabet based not only on which sound classes exhibit similar behaviors, but also on whether the value for particular feature changes in a given grammar. For instance, in Lumasaaba, the learned partition for [voice] separates vowels from voiced and voiceless consonants. Since the value of [voice] does not change for l even when it surfaces as /o/, isolating l as a singleton is not strictly necessary for sample consistency—yet doing so substantially improves *overall* generalization.

The imposed Φ sets were designed to approximate linguistically motivated natural class divisions. However, there is no guarantee that these combinations are empirically optimal. It is a reasonable assumption that comparable or even better generalizations could be achieved with fewer features or with a smaller value of k for a given dataset.

The purpose of this section is therefore not to identify ideal featural combinations, but rather to demonstrate that the choice of Φ indeed affects learning outcomes. Different partitions of Σ_{seg} can either improve or degrade the quality of generalization, even when overall machine complexity remains unchanged. These findings further motivate the incorporation of additional inductive biases or some information, whether learned from data or specified in advance, that could guide the learner toward more linguistically appropriate partitions and thereby improve performance.

The transition-level and out-of-vocabulary analyses show that Serbo-Croatian is

a particularly informative case. The learner can produce plausible outputs for constructed test forms, but the transition-level match with the gold transducer remains low, and many matching transitions occur in unattested regions of the state space. We therefore use Serbo-Croatian as a limited held-out diagnostic within the textbook-style datasets.

6.6.3 Held-out evaluation in Serbo-Croatian

The preceding analyses evaluated the learned transducers in three ways: by testing whether they reproduce the observed input–output pairs, by comparing the recomposed learned transducer DFT_L against the gold transducer DFT_G at the transition level, and by probing the learned mappings with a small set of manually constructed out-of-vocabulary forms. These diagnostics were chosen due to the sparsity of the textbook-style datasets, where the goal is not to estimate a conventional test-set error rate, but to determine what kind of generalization is good enough given the limited evidence available to the learner.

The Serbo-Croatian results raise a further question. On the one hand, the learned transducer appears to generalize beyond the training sample. The learned grammar mapped the manually constructed out-of-vocabulary input to expected outputs. On the other hand, the transition-level comparison provides only limited support for this conclusion. In Serbo-Croatian, DFT_L matches only a small proportion of the transitions in DFT_G under the learner-selected feature specification, with higher but still incomplete agreement under the imposed-Φ condition. Moreover, almost all matching transitions correspond to unattested k-gram contexts in the original dataset. This suggests that large regions of the transducer remain underdetermined by the available sample. Consequently, success on a small set of out-of-vocabulary probes does not by itself show that the learner has recovered the intended mapping. We therefore ask whether the Serbo-Croatian generalization is stable under a more systematic held-out

evaluation.

To address this question, we conducted a 5-fold cross-validation experiment. The dataset is highly imbalanced: of the 132 input–output pairs, 113 are identity mappings and only 19 exhibit one of the two target alternations. The folds were therefore constructed so that each training and held-out split contained both identity and non-identity mappings. In each fold, one subset was held out for testing and the remaining four subsets were used for training. Since 132 does not divide evenly into five folds, held-out sets contained between 25 and 27 items.

We ran two versions of the experiment. In the first condition, the learner used the feature combinations selected in the original Serbo-Croatian experiments. In the second condition, we used the imposed feature set, which produced somewhat better transition-level agreement with the gold transducer in the preceding analysis. The feature specifications that differ between the two conditions are summarized in Table 6.13; the full feature sets are provided in Appendix C, Table C.17.

Learner-selected	**Manually imposed**
[**son**, cons]	[**son**, lat]
[**cont**, cons]	[**cont**, lat]
[**delrel**, cons]	[**delrel**, lat]
[**approx**, trill]	[**approx**, lat]
[**trill**, approx]	[**trill**, lat]
[**nasal**, cons]	[**nasal**, lat]
[**voice**, cons]	[**voice**, lat]

TABLE 6.13: Feature specifications that differed between the learner-selected and manually imposed Serbo-Croatian conditions. The base feature is shown in bold. All other base features used the same feature specification in both conditions.

The pair-level cross-validation results are presented in Table 6.14 below. Accuracy was calculated separately for each fold, processing direction, and feature-specification

condition. A held-out UR–SR pair was counted as correct only if the recomposed segmental transducer returned the correct segmental output. Thus, an error at any feature projection was counted as an incorrect prediction for the whole pair.

Spec.	Dir.	Fold	Train	Test	Non-id.	Learner err.	Id. err.	Non-id. err.	Learner acc.	Baseline acc.
Feature combinations selected by the learner										
learned	L2R	1	105	27	4	4	2	2	85.19	85.19
		2	105	27	4	5	3	2	81.48	85.19
		3	105	27	4	12	11	1	55.56	85.19
		4	106	26	4	4	4	0	84.62	84.62
		5	107	25	3	5	3	2	80.00	88.00
	R2L	1	105	27	4	4	2	2	85.19	85.19
		2	105	27	4	5	3	2	81.48	85.19
		3	105	27	4	12	11	1	55.56	85.19
		4	106	26	4	2	2	0	92.31	84.62
		5	107	25	3	5	3	2	80.00	88.00
Manually imposed phonological feature combinations										
manual	L2R	1	105	27	4	4	2	2	85.19	85.19
		2	105	27	4	6	3	3	77.78	85.19
		3	105	27	4	11	9	2	59.26	85.19
		4	106	26	4	4	4	0	84.62	84.62
		5	107	25	3	5	3	2	80.00	88.00
	R2L	1	105	27	4	4	2	2	85.19	85.19
		2	105	27	4	5	3	2	81.48	85.19
		3	105	27	4	11	9	2	59.26	85.19
		4	106	26	4	2	2	0	92.31	84.62
		5	107	25	3	5	3	2	80.00	88.00

TABLE 6.14: Pair-level 5-fold cross-validation results for Serbo-Croatian. Learner errors count held-out segmental pairs for which at least one feature projection was incorrectly predicted. Id. err. and Non-id. err. split those errors by whether the held-out UR–SR pair was identical or non-identical. The identity baseline treats all identity pairs as correct and all non-identity pairs as incorrect.

We also compare the learned transducers against an identity baseline. This baseline maps every held-out form faithfully from UR to SR. All identity pairs are therefore counted as correct, while all non-identity pairs are counted as incorrect. Given the imbalance of the Serbo-Croatian dataset, this is a strong baseline, as a learner can obtain high accuracy simply by preserving the input. Thus, this baseline provides a useful reference point for evaluating whether LCD-SEAL improves on default faithfulness.

The learned mappings do not consistently outperform this baseline. Under the learner-selected feature specifications, mean accuracy is 77.37% in the L2R direction and 78.90% in the R2L direction, compared to an identity-baseline accuracy of 85.64%.

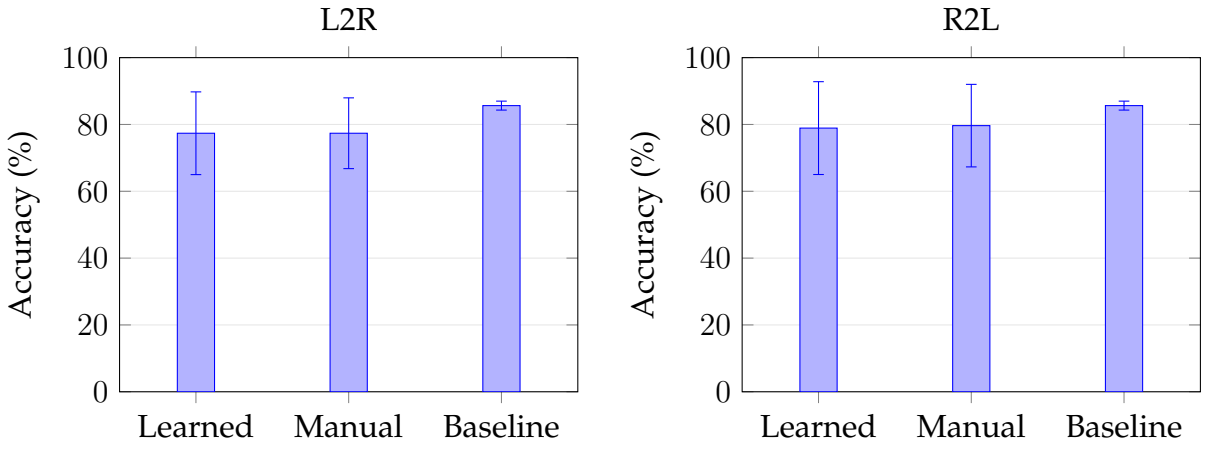


FIGURE 6.2: Mean pair-level held-out accuracy across five Serbo-Croatian cross-validation folds. Error bars show standard deviation across folds. The identity baseline predicts UR = SR, correctly mapping all identity pairs and failing on all non-identity pairs.

Under the manually imposed feature specifications, mean accuracy is nearly identical: 77.37% in L2R and 79.65% in R2L, again below the identity baseline. These results are summarized in Figure 6.2, which shows mean held-out accuracy across the five folds for each condition and direction.

Although mean accuracy remains below the baseline, the fold-level results are not uniform. In some folds, most clearly in Fold 4, R2L direction, the learner outperforms the identity baseline. Moreover, LCD-SEAL performs better than the identity baseline on non-identity UR–SR pairs: across folds and conditions, it correctly generates at least some outputs for alternating URs, whereas the identity baseline fails on all such cases. The failed test pairs and the features that contributed to these errors are listed in Appendix C, Table C.19.

Many errors occur on identity pairs. This means that the learned mapping sometimes introduces changes where the correct output should preserve the input. For example, in Fold 3, in the feature-learned conditions the learner produced 12 total errors, 11 of which occur on identity pairs. The corresponding manually imposed condition is only slightly better, with 11 total errors, 9 of which occur on identity pairs. This shows that the learner also assigns feature-level mappings that incorrectly alter

otherwise faithful forms.

At the same time, the learner does capture some non-identity mappings. On average, over 50% of non-identical surface forms are correctly inferred, which is never the case for the baseline learner. Interestingly, although the imposed feature sets Φ were designed to induce more phonologically coherent natural classes, and although they improved transition-level agreement with the gold transducer, they do not yield a clear improvement in held-out pair-level accuracy.

The Serbo-Croatian patterns are therefore not recovered uniformly across held-out folds, even when the corresponding training sets contain examples of both a-epenthesis and final l-vocalization. The failures do not reflect a simple absence of evidence for the relevant alternations. Rather, they reflect the interaction between sparse evidence and the learning procedure. LCD-SEAL searches for an assignment that is consistent with the observed training sample, but such an assignment is not guaranteed to be the best generalization for unseen forms. A transducer may therefore fit the training data while remaining too specific to the sample, or too aggressive on identity mappings.

This interpretation is supported by the distribution of errors across folds and directions. The learner often succeeds on many held-out pairs, but its errors are unevenly distributed, with particularly low accuracy in Fold 3 across all conditions. The fact that the learner-selected and imposed feature specifications exhibit the same broad pattern suggests that the problem cannot be reduced to the choice of the Φ feature sets. Rather, the main issue appears to involve selecting the most generalizable assignment. Future work should therefore investigate additional biases on the LCD-SEAL search procedure, so that the selected assignment is not merely the first compatible one, but the one that is phonologically most plausible.

Despite the limited size of the Serbo-Croatian dataset, the cross-validation study shows that LCD-SEAL can achieve nearly 80% held-out accuracy from only 107 training examples. This result is still informative, especially because the learner correctly

predicts the surface forms for an average of approximately 50% of the non-identity URs. Thus, the learner is not merely reproducing the dominant identity pattern in the dataset. Importantly, it also captures part of the alternation pattern.

This result is particularly notable in comparison with the synthetic-data Chukchi case study discussed in the previous chapter. Chukchi also involved epenthesis, but LCD-SEAL required approximately 650 examples to learn the gold transducer, despite the fact that the artificial dataset contained a highly restricted alphabet of only seven phonemes. By contrast, the Serbo-Croatian dataset contains only a fraction of that characteristic sample size and has a substantially larger phonemic inventory, with 22 phonemes. The Serbo-Croatian results therefore show that LCD-SEAL can extract non-trivial generalizations from considerably sparser and more naturalistic data.

At the same time, this experiment also identifies a clear weakness of the current version of LCD-SEAL. Although the learner reaches relatively high overall accuracy, it does not consistently outperform the identity baseline and still makes errors on both alternating and identity forms. This suggests that the current search procedure can find training-consistent mappings, but it does not always select the generalization that extends best to unseen data. This limitation points to a concrete direction for future work, which should focus on developing stronger selection biases so that LCD-SEAL prefers not only consistent hypotheses, but also ones that are phonologically plausible and generalize more robustly.

The Serbo-Croatian cross-validation study provides a held-out diagnostic within the textbook-style setting, but the dataset remains small and the alternations are weakly represented relative to identity mappings. To test SEAL under a larger and more explicitly acquisition-oriented condition, we now turn to a corpus-derived Polish dataset based on CHILDES.

6.6.4 Corpus-based cross-validation: Polish CHILDES

The experiments reported so far in this chapter use naturalistic, textbook-style datasets. These datasets are substantially more realistic than the synthetic datasets discussed in Chapter 4, but they remain limited in two respects. First, they contain relatively small numbers of lexical items. Second, they are constructed specifically to illustrate the phonological processes of interest, and therefore do not fully reflect the frequency distribution of forms in the input available to children. In this section, we therefore introduce an additional experiment based on Polish child-directed speech from CHILDES. The goal of this experiment is to test whether FxF(SEAL) can learn the same Polish processes examined in Chapter 4, namely word-final devoicing and ɔ-raising, from a *natural* dataset.

The dataset was constructed from the Polish Frequency List of Child-Directed Speech, available through CHILDES. The corpus is based on utterances produced by parents, grandparents, experimenters, and other speakers older than eight-year-old in the presence of children. Although not all utterances are strictly child-directed, they represent speech to which children were exposed. The frequency list combines material from seven Polish corpora, with the oldest data coming from the 1940s and the newest from the 2000s. This makes the dataset particularly useful for our purpose, since it approximates the lexical input available to a child, with *word types* ordered by frequency.

For the experiment, we selected approximately the first 5,000 most frequent word types from the frequency list, excluding function words, since these do not provide relevant evidence for the two phonological processes targeted here. The resulting dataset contains 5,003 underlying–surface pairs. The final number is slightly larger than 5,000 because we chose not to break morphological paradigms. Of these 5,003 pairs, 4,872 are identity mappings and 131 are non-identity mappings. The dataset is therefore highly imbalanced, such that only 2.6% of the data provides evidence for either word-final devoicing or ɔ-raising.

As in the other datasets discussed in this chapter, the underlying representations were created manually. However, in this experiment the underlying forms were reconstructed only with respect to the two processes under investigation. The URs are then limited in the sense that they do not attempt to account for other phonological alternations in Polish, such as velar palatalization or regressive voicing assimilation. This is an intentional restriction. The purpose of the experiment is not to construct a complete phonological grammar of Polish, but to determine whether the same learner that succeeded on idealized synthetic data can recover the relevant mappings from a natural, frequency-ordered sample. This setup therefore allows for a more direct comparison with the Polish experiments in Chapter 4.

A small number of forms required special treatment. Forms that met the structural description for ɔ-raising but did not exhibit the change were removed from the dataset. For example, kɔlɔr does not surface as *kɔlur but as identity, despite satisfying the relevant environment. Since the aim of the experiment is to test learning of the productive mapping rather than exceptions, such cases were excluded. In addition, two forms occurred with more than one surface pronunciation. The form /uwɔzˌ/ appears both as [uwuʂ] and as [uwɔʂ], and /zawɔzˌ/ likewise appears both as [zawuʂ] and [zawɔʂ]. For the purposes of the present experiment, we retained only the variants exhibiting ɔ-raising. This choice is justified by the historical breadth of the corpus: because the data span several decades, including material from the 1940s through the early 2000s, the raised variants can plausibly be treated as part of the phonological evidence represented in the corpus, even if they may be less acceptable in present-day standard Polish.

The evaluation differs from the one used for the smaller textbook-style datasets. In those experiments, the samples were often too small to support a heldout test set without substantially reducing the evidence available to the learner. In the Polish CHILDES experiment, by contrast, the dataset is large enough to permit cross-validation. We therefore used 10-fold cross-validation. In each fold, approximately

90% of the data formed the training pool and approximately 10% formed the heldout test set. Across the full experiment, every UR–SR pair occurred in a heldout test set exactly once.

Because the dataset is highly imbalanced, folds were divided by pair type. Identity mappings and non-identity mappings were first separated, and each fold received approximately the same proportion of both types. This is important because the non-identity mappings provide the direct evidence for the two alternations. Without this stratification, some heldout folds could contain too few alternating forms, making the evaluation less informative.

After the heldout test set was selected for a given fold, the remaining training items were returned to their original order in the frequency list. This means that the training pool was not randomly shuffled. Instead, the learner was exposed to increasingly less frequent forms over time. This is crucial to the interpretation of the experiment because the learner is not simply trained on a random subset of 4,500 examples, but on growing frequency-ordered items of the available training data.

Within each fold, the learner was trained incrementally. To reduce runtime, we first identified the successful input-feature configuration for each fold; this configuration turned out to be the same across all folds for all $\phi \in F$. We then conducted the incremental-learning experiment using these fixed feature sets Φ. The heldout test set remained fixed, while the training sample increased in size. Two batch sizes were tested. In the batch-size-10 condition, the learner was evaluated after seeing the first training pair, then after the first 10 pairs, then after the first 20 pairs, and so on. In the batch-size-100 condition, the learner was evaluated after the first training pair, then after the first 100 pairs, then after the first 200 pairs, and so on. These batches were cumulative: the 200-pair training condition contained the first 200 forms in the frequency-ordered training pool, not a separate block of 200 new forms.

For each fold, two quantities were recorded: the total number of training pairs required before the learner passed the heldout test set, and the number of non-identity

pairs contained within that successful training set. The second measure is particularly important, since the informative evidence for the alternations is sparse. A large training prefix may contain relatively few alternating forms, depending on where those forms occur in the frequency-ordered corpus.

The learner was evaluated feature by feature, as in the other experiments in this chapter. For each output feature, SEAL constructed a feature-level transducer. Most features could be learned with the initial identity assumption. However, the features directly involved in the two target processes required larger Φ sets, more memory, and consequently larger machines. The relevant settings for the *insufficient* features are presented in Table 6.15. Finally the experiments in the synthetic datasets clearly demonstrated that R2L direction is more successful in reducing the sample size, we chose this direction as a default. Only if R2L was not successful, the learner was directed to use L2R.

Output feature	**Input-feature configuration Φ**	k
VOICE	[VOICE, SON]	2
HIGH	[HIGH, NASAL, ROUND, TAP, VOICE]	3
TENSE	[TENSE, NASAL, ROUND, TAP, VOICE]	3

TABLE 6.15: Successful input-feature configurations for the Polish CHILDES experiment.

A fold was counted as successful only when all relevant feature-level transducers produced the correct outputs on the heldout test set. The heldout test set was never used for training.

Training size required for 100% accuracy held-out performance

Table 6.16 reports the number of frequency-ordered training pairs required to reach 100% held-out accuracy in each fold. The table also reports the number of non-identity pairs included in the train batches.

Fold	Train	Test	Batch size 10		Batch size 100	
			Pairs needed	Non-id. needed	Pairs needed	Non-id. needed
1	4501	502	1410	33	1500	34
2	4502	501	110	2	200	3
3	4503	500	1480	36	1500	36
4	4503	500	110	2	200	3
5	4503	500	2210	51	2300	53
6	4503	500	970	17	1000	18
7	4503	500	240	5	300	6
8	4503	500	290	5	300	5
9	4503	500	1420	34	1500	35
10	4503	500	980	18	1000	19

TABLE 6.16: Polish incremental 10-fold cross-validation results. Each fold used a frequency-ordered training pool and a fixed heldout test set. *Pairs needed* reports the smallest cumulative training prefix that passed the heldout test set. *Non-id. needed* reports how many non-identical UR-SR pairs occurred in that successful prefix.

The results show that the learner did not require the full training pool in any fold. With batch size 10, successful generalization occurred after as few as 110 training pairs in folds 2 and 4, and after as many as 2,210 training pairs in fold 5. The number of non-identity pairs contained in the successful prefix ranged from only 2 to 51. Thus, although each fold had access to approximately 4,500 training examples, SEAL typically succeeded after seeing a substantially smaller subset of the most frequent available forms.

The batch-size-100 condition shows the same overall pattern, but at a coarser level. Since the learner was evaluated only at intervals of 100 examples, the detected success points are generally slightly later than in the batch-size-10 condition. For example, fold 1 succeeded after 1,410 examples in the batch-size-10 condition but after 1,500 examples in the batch-size-100 condition. Similarly, folds 2 and 4 succeeded after 110 examples with batch size 10 but after 200 examples with batch size 100. These differences reflect the granularity of evaluation rather than a substantive difference in learning behavior.

There is nevertheless considerable variation across folds. Some folds passed after

exposure to a very small number of training pairs, while others required more than 1,000 examples. This variation is expected under the experimental design. Since the training data are frequency ordered, the crucial factor is not simply the size of the training set, but the point at which informative non-identity examples occur in the remaining training pool. If the relevant alternating forms appear early, the learner succeeds with few total examples. If they appear later, a larger amount of data is required.

The contrast between folds 4 and 5 illustrates why the number and type of non-identity examples matter. Fold 4 was learned after only two non-identity training examples: viɛd͡ʒ ↦ viɛt͡ɕ and rɔb ↦ rup. These two examples provided evidence for both target alternations: word-final devoicing and ɔ-raising. In this fold, the held-out non-identity forms appear to require no substantially broader featural generalization than what was already supported by these early examples. As a result, the learner was able to pass the held-out set after a very small training prefix.

Fold 5 presents a slightly different case. Although the learner also encountered early evidence for the target mappings, the held-out set included a wider range of ɔ-raising contexts, including liquid and glide environments in addition to obstruent-triggered cases. These contexts were not sufficiently supported by the earliest non-identity examples in the training prefix. Consequently, the learner required a much larger amount of data before it encountered the relevant featural evidence. The full list of non-identity training forms for fold 5 is provided in Appendix C, Table C.18. From this table it can be inferred that that once examples exhibiting ɔ-raising in the relevant liquid l and glide w contexts appeared in the training sample, the learner succeeded on the held-out set, and the learning procedure stopped.

This pattern is phonologically informative. Obstruents form a relatively large and coherent natural class, and therefore evidence from one obstruent context can support generalization to other obstruent contexts. By contrast, the liquids l and r, and glides w and j, are not mutually recoverable from a single example in the same way. The

variation across folds therefore reflects not only the number of alternating examples, but also their featural distribution.

The detailed held-out UR–SR pairs that remained problematic at intermediate points in the learning process are reported in Appendix C, Tables C.20 and C.20, for the batch-size-10 and batch-size-100 conditions, respectively. These tables list, for each fold, the problematic test-set forms, whether they are identity or non-identity mappings, the features that contributed to the error, and the size of training sample at which the right surface form was predicted. In all cases, the errors involved one or more of the three insufficient features: [voice], [high], and [tense]. Since the learned grammar did not involve any deletions or insertions at segmental level, all remaining features could easily be inferred by assuming identity.

Incremental accuracy and comparison with the identity baseline

The preceding analysis identifies the point at which each fold first reached 100% held-out accuracy. However, it does not show how performance improved before that point. To track the gradual improvement of SEAL, we additionally measured held-out accuracy across incremental training checkpoints for both batch sizes. The batch-size-100 results are given in Table 6.17. The finer-grained batch-size-10 results are shown in Table 6.18. This condition provides a more precise view of the points at which individual folds reach 100% accuracy, especially early in the training sequence.

Figure 6.3 summarizes the batch-size-100 condition and compares the learner against an identity baseline. Because the dataset is overwhelmingly composed of identity mappings, a learner that simply maps every UR faithfully to itself reaches 97.38% accuracy.

The comparison with the identity baseline is particularly informative. After a single training pair, mean held-out accuracy is already 95.08%, reflecting the dominance of identity mappings in the dataset. After 100 training pairs, mean accuracy rises to 97%, still slightly below the identity baseline. After 200 training pairs, however,

Train	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
1	94.2	96.2	95.4	95.4	95.4	94.2	95.2	94.2	95.6	95.0	95.1	0.6
100	96.8	98.4	97.4	97.4	97.2	96.2	97.2	96.6	95.6	97.2	97.0	0.8
200	99.6	100.0	99.8	100.0	99.6	99.6	99.8	99.6	99.6	99.2	99.7	0.2
300	99.6	–	99.8	–	99.8	99.6	100.0	100.0	99.6	99.2	99.7	0.3
400	99.6	–	99.8	–	99.8	99.6	–	–	99.6	99.2	99.6	0.2
500	99.6	–	99.8	–	99.8	99.8	–	–	99.6	99.2	99.6	0.2
1000	99.6	–	99.8	–	99.8	100.0	–	–	99.6	100.0	99.8	0.2
1500	100.0	–	100.0	–	99.8	–	–	–	100.0	–	100.0	0.1
2300	–	–	–	–	100.0	–	–	–	–	–	100.0	–

TABLE 6.17: Growth in held-out pair-level accuracy across incremental training checkpoints for Polish in the fixed-specification R2L identity-learner condition with batch size 100. Values are percentages. Dashes indicate that the fold had already reached 100% accuracy and stopped before that checkpoint.

Train	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Mean	SD
1	94.2	96.2	95.4	95.4	95.4	94.2	95.2	94.2	95.6	95.0	95.1	0.7
10	94.2	96.2	95.4	95.4	95.4	94.2	95.2	94.2	95.6	95.0	95.1	0.7
80	96.8	98.4	97.4	97.4	97.2	96.2	97.2	96.6	95.6	97.2	97.0	0.8
100	96.8	98.4	97.4	97.4	97.2	96.2	97.2	96.6	95.6	97.2	97.0	0.8
110	99.6	100.0	99.6	100.0	99.6	99.6	97.2	96.6	99.4	99.2	99.1	1.2
200	99.6	–	99.6	–	99.6	99.6	99.8	96.6	99.4	99.2	99.2	1.1
240	99.6	–	99.8	–	99.8	99.8	100.0	98.4	99.6	99.2	99.5	0.5
290	99.6	–	99.8	–	99.8	99.8	–	100.0	99.6	99.2	99.7	0.3
970	99.6	–	99.8	–	99.8	100.0	–	–	99.6	99.2	99.7	0.3
980	99.6	–	99.8	–	99.8	–	–	–	99.6	100.0	99.8	0.2
1410	100.0	–	99.8	–	99.8	–	–	–	99.6	–	99.8	0.2
1420	–	–	99.8	–	99.8	–	–	–	100.0	–	99.9	0.1
1480	–	–	100.0	–	99.8	–	–	–	–	–	99.9	0.1
2210	–	–	–	–	100.0	–	–	–	–	–	100.0	–

TABLE 6.18: Growth in held-out pair-level accuracy across incremental training checkpoints for Polish in the fixed-specification R2L identity-learner condition with batch size 10. Values are percentages. Dashes indicate that the fold had already reached 100% accuracy and stopped before that checkpoint.

mean accuracy reaches 99.34%, surpassing the baseline. Accuracy then continues to improve more gradually, reaching 99.78% after 300 training pairs, 99.88% after 1,000 training pairs, 99.98% after 1,500 training pairs, and finally 100% after 2,300 training pairs. Thus, once enough informative examples are encountered, the learner improves beyond identity and begins to capture the non-identity alternations.

This result contrasts sharply with the Serbo-Croatian cross-validation experiment. In Serbo-Croatian, the learner achieved relatively high held-out accuracy, but it did not consistently outperform the identity baseline. In the Polish CHILDES experiment,

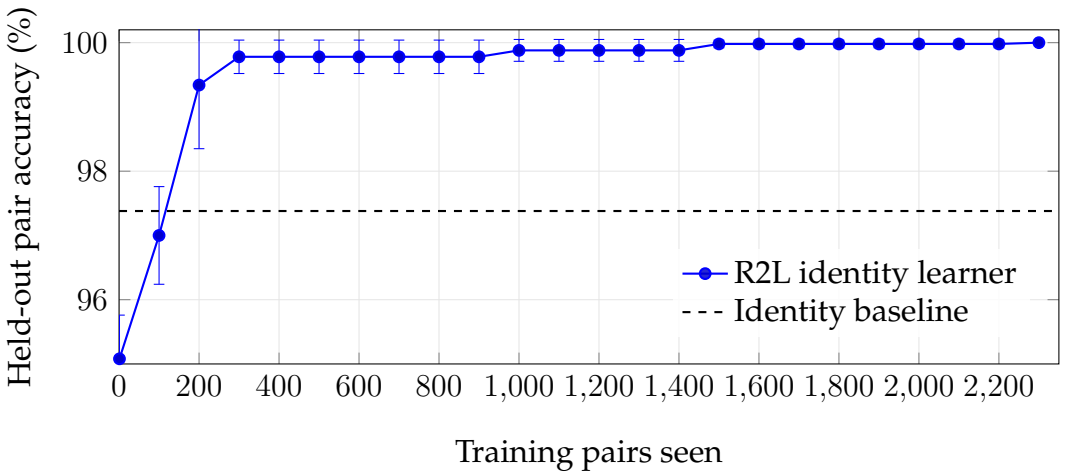


FIGURE 6.3: Mean held-out pair-level accuracy across ten Polish cross-validation folds in the fixed-specification R2L identity-learner condition. Error bars show standard deviation across folds. The dashed line marks the identity-baseline accuracy of 97.38%, obtained by predicting all UR–SR pairs as identity mappings.

the identity baseline is even stronger, yet the learner exceeds it after approximately 200 frequency-ordered training pairs. This suggests that the Polish mappings are supported by the distribution of evidence in the CHILDES sample in a way that the Serbo-Croatian alternations were not consistently supported by the much smaller textbook-style dataset.

The CHILDES results are also directly comparable to the synthetic Polish experiments in Chapter 4. In that chapter, PRD-SEAL required 102 examples to learn the same combination of word-final devoicing and ɔ-raising, using an input alphabet of solely seven phonemes. In the present experiment, folds 2 and 4 required 110 training examples in the batch-size-10 condition, despite the fact that the Polish CHILDES dataset uses a substantially larger phonemic inventory (36 phonemes). This comparison is a great example of the benefits of featural representations: the same approximate amount of evidence can support learning across very different alphabet sizes, provided that the relevant featural contrasts are available.

Overall, the Polish CHILDES experiment supports the conclusion that SEAL can learn the targeted Polish mappings from naturalistic, frequency-ordered input. The

learner does not require exposure to the entire corpus, nor does it require a large number of non-identity examples in every fold. In the finer-grained batch-size-10 condition, successful held-out performance required between 2 and 51 alternating forms, depending on which contexts were represented in the early training sample. At the same time, the results show that the distribution of those forms does matter. Sparse evidence is sufficient for successful inference but only when it contains the featural contrasts needed to support the relevant generalization. In this respect, the CHILDES experiment extends the earlier synthetic Polish results by showing that the same mappings can be recovered under significantly more realistic conditions, where the learner is exposed to a frequency-ordered sample resembling the type-frequency structure of child-directed input.

6.7 Discussion

The results presented in this chapter demonstrate that FxF SEAL successfully learns morphophonological mappings from sparse, textbook-style datasets. Each function was evaluated both at the featural level and after synthesis at a segmental level. In all five textbook-style cases, the learner inferred transducers that were consistent with the observed input–output pairs. While it may be tempting to interpret this success as mere memorization of observed paths, this interpretation does not hold.

A prefix-tree transducer provides the canonical memorization baseline. It encodes exactly the observed input–output pairs, remains acyclic, and grows proportionally with the number of distinct prefixes in the data. In contrast, SEAL's learned transducers do contain cycles.

Interestingly, SEAL does not perform explicit state merging. Instead, compression arises from the structural constraints imposed by the k-ISL hypothesis space and incorporation of the FxF system. Each featural learner assigns outputs to placeholders shared across multiple contexts. Because placeholders correspond to recurrent k-local

configurations, identical assignments are reused across distinct input paths.

In grammatical inference more broadly, cycles are the hallmark of generalization beyond memorization as they indicate that the same state can be revisited under different inputs, which enables productive and potentially unbounded behavior. SEAL exhibits this property not through explicit merging, but through hypothesis restriction and structured reuse of contexts. Thus, the full-sample results show that SEAL is not simply storing the training data.

At the same time, the held-out evaluations refine this picture. They show that sample consistency and cyclic structure may not, by themselves, be sufficient to guarantee robust generalization to withheld attested forms. The Serbo-Croatian cross-validation study provides the clearest example of this point. Although the learner achieved relatively high held-out accuracy, it did not consistently outperform the identity baseline. In that dataset, identity mappings dominate, and the relevant alternations are represented by relatively few examples. As a result, high pair-level accuracy can be achieved without reliably improving on a strategy that simply predicts identity.

The Polish CHILDES experiment shows that this limitation is not simply a consequence of learning from sparse or imbalanced data. The Polish dataset is also highly identity-dominant, and the identity baseline is even stronger than in Serbo-Croatian. Nevertheless, SEAL surpasses this baseline after as few as 110 frequency-ordered training pairs and eventually reaches perfect held-out accuracy. Together, the two cross-validation studies suggest that what matters is not only the number of examples or the proportion of alternating forms, but also the nature of the alternations and the featural contrasts available in the sample. In Serbo-Croatian, the textbook-style dataset is large enough to permit held-out evaluation, but the relevant alternations may not be consistently supported across folds. Moreover, the targeted grammar involves segment insertion and the vocalization of a highly specific segment, l, rather than a broad natural class that could support wider featural generalization. In Polish, by contrast,

the larger frequency-ordered CHILDES sample provides enough distributional evidence for the learner to recover the target mappings. The fact that part of the Polish grammar targets the broad natural class of obstruents may further facilitate learning relative to the more narrowly specified Serbo-Croatian alternations.

An indirect advantage of feature-based learning, observed both in SOSFIA and SEAL, is that it allows partial disentanglement of interacting processes. In FxF SOSFIA, learning Polish o-raising and word-final devoicing revealed that different features required different memory depths: [voice] required $k = 2$, whereas [high] and [tense] required $k = 3$, and each targeted distinct elements and environments. Although this decomposition is not explicitly encoded, it may emerge naturally from feature-wise learning.

A similar pattern appears in Lumasaaba and Serbo-Croatian. Different features capture different aspects of the phonological system, and the required memory window varies accordingly. For example, in Lumasaaba only features [coronal], [distributed], [strident], [anterior], [front] and [back] divided the class of velar stops and front vowels from the remainder of the inventory. These are precisely the features that are active in velar palatalization when k surfaces as c. This suggests that feature-wise learning may provide a window into latent structural decomposition of complex phonological processes.

A recurring observation throughout the experiments is that the partition of the segmental alphabet Σ induced by Φ can play an important role in determining generalization quality. When the induced partition aligns more with phonologically motivated natural classes, generalization seems to be closer to the gold standard. This point is visible in the transition-level results, where imposed featural combinations improve performance for Lumasaaba and Serbo-Croatian. It is also visible in the held-out evaluations.

SEAL is fundamentally a data-informed learner. It constructs hypothesis based only on observed paths and defaults to identity on unseen ones. From an efficiency

perspective, this assumption is not unreasonable as preserving the input in the absence of evidence is less expensive computationally than changing it. The Polish CHILDES results show that this bias contributes to successful learning: even with a very strong identity baseline, SEAL can recover non-identity mappings once the relevant evidence is present. However, the Serbo-Croatian results show that identity can also be a difficult baseline to outperform when alternations are sparse, inconsistently represented, and the targeted grammar involves insertion/deletion of an entire segment.

However, this assumption is not without consequences. Because identity is enforced at the transition level rather than on paths level, the learner does not verify whether outputting a value along an unseen path is globally coherent. As seen in, for example, Samoan, this can produce implausible feature sequences, including mismatched feature-value counts across features. This shows that the same bias that supports data-efficient learning can also generate systematic errors in sparse data conditions. Future work should explore mechanisms for constraining identity defaults, perhaps through path-level consistency checks. For example, if an output is withheld at some level, the identity assumption should take this absence into account, rather than ignoring it and, as a consequence, producing missing or inconsistent feature values elsewhere in the representation.

In some cases, additional data could help improve those issues by creating more linguistically faithful classes, as observed in Lumasaaba and Serbo-Croatian. In other cases, there is no possible featural combination that would group certain sounds together for particular feature. For example, in Samoan, no feature set Φ can group {i,e} as required for [high] because one phoneme is high and the other is not. Correct generalization therefore requires exposure to all logically relevant substrings (ii, ee, ie, ei). In such cases, the limitation is not the quality of learned feature combinations, but representational insufficiency.

The two modes of SEAL, the local push-right (`PRD`) and global length-constrained (`LCD`), reflect different inductive assumptions. `PRD-SEAL` is phonologically motivated

in a way that it enables output withholding until contextual information becomes available only when *conflict* arises. It performs well for learning single processes and interactions that do not involve epenthesis or deletion.

`LCD-SEAL` relaxes this constraint by allowing empty transitions and non-identity outputs in various positions, as long as the individual outputs can be consistently distributed across the placeholders. This flexibility is necessary for deletion and epenthesis, but it also introduces more ambiguity, since multiple consistent assignments may exist and nothing in the current system determines which one is better. The Serbo-Croatian cross-validation study makes this limitation particularly clear. LCD-SEAL can extract non-trivial generalizations from sparse, naturalistic data, but the current search procedure does not always select the hypothesis that generalizes best to unseen forms. Future work may incorporate preferences such as minimizing output length, favoring singleton assignments where possible, or otherwise preferring hypotheses that are both consistent and phonologically plausible.

The feature system adopted here from Hayes (2009) is relatively rich allowing for binary and ternary values. Alternative systems may also contribute to different results. For example, Zsiga (2013) allows also for unary features which, in principle, could reduce representational complexity because it could further restrict the size of feature alphabets. For example, when learning vowel nasalization in English, the only possible featural alphabet elements would be $\left[\begin{smallmatrix}+\text{cons}\\ \text{nasal}\end{smallmatrix}\right]$ or $\left[\begin{smallmatrix}-\text{cons}\\ \text{nasal}\end{smallmatrix}\right]$. Under this system, nasality either exists or not so no natural class containing elements specified for [- nasal] can emerge.

A second example comes from place assimilation. Zsiga advocates the usefulness of autosegmental representations and dependency-based feature geometry, in which features are hierarchically organized rather than listed flatly. In such a system, segments are represented as bundles of nodes arranged in a structured geometry, with suprasegmental organization explicitly encoded. Consequently, for example, place

features are grouped under a Place node, which itself is embedded within a larger hierarchical structure. Under this representation, place assimilation, as in Koasati, can be modeled as the spreading (of filling) of a Place node rather than as multiple independent feature changes as it was the case in a flat feature system such as Hayes'.

Of course, such systems introduce their own challenges. Because vowels and consonants are represented differently in feature geometry, processes like velar palatalization in Lumasaaba may require additional assumptions. However, no feature system is without trade-offs. Even in the present implementation, adjustments were necessary, for example, assigning a default value of 0 to consonantal features for vowels or underspecified segments.

Nonetheless, the hierarchical organization of features in a geometry-based system may offer important benefits. By structurally constraining how features can combine, it restricts the hypothesis space and may make the search for a featural set Φ more efficient. Thus, while the richer system adopted here provides expressive power, we expect that alternative systems might yield different generalizations.

Experiments conducted in this chapter also raise a broader question, particularly what is the *right* phonological rule? Should Koasati assimilation target only the underspecified nasal, or all nasals? If the data show alternations only for one segment, is it preferable to generalize narrowly or broadly?

Similarly, in Samoan, some predictive "errors" arise on unattested consonant clusters. If the training data contain no such clusters, should these predictions count as failures? Incorporating phonotactic constraint, either learned or assumed, may provide a principled way to distinguish genuine overgeneration from predictions in unattested contexts.

Questions of this type are central in the phonological learning literature. Work on productivity and analogy has shown that learners often extend patterns beyond the strictly attested cases, but that such extensions are guided by structural similarity and natural class structure rather than unrestricted generalization (Albright and

Hayes, 2002, 2003). This connects to broader questions about how learners arrive at generalizations that are powerful enough to be productive while remaining appropriately constrained (Jarosz, 2011). Future research could utilize SEAL on child-directed speech datasets and evaluate whether the resulting generalizations align with known patterns of children's productivity, including natural-class–based extensions.

6.8 Conclusion

The experiments in this chapter show both the strengths and the boundaries of bounded k-ISL learning under featural decomposition. Within the imposed constraints on memory ($k \leq 4$) and feature-set size ($|\Phi| \leq 5$), SEAL can infer compact, cyclic transducers that, to some extent, generalize beyond the observed data while remaining consistent with the training sample. These results support the view that much of morphophonology can be captured through strictly local mechanisms when the representational space is appropriately structured.

At the same time, the chapter also shows that training-sample consistency is not equivalent to robust generalization. Transition-level comparison, constructed OOV forms, and held-out evaluation provide different views of what the learner has acquired. The cross-validation results sharpen this point. Serbo-Croatian shows that fitting a sparse textbook-style sample does not by itself guarantee robust prediction of randomly selected held-out forms, especially when the dataset is dominated by identity mappings and the grammar targets insertion. Polish, however, shows that SEAL can recover the target mappings from frequency-ordered corpus-derived input and can outperform a strong identity baseline once the relevant featural contrasts are represented in the training sample.

Thus, the central conclusion is not simply that more data improves learning. Rather, the results suggest that successful generalization depends on the interaction between data distribution, featural representation, and hypothesis selection. Sparse evidence

can be sufficient, but only when it supports the feature partitions needed for the target mapping. When those partitions are poorly supported, as in parts of the Serbo-Croatian and Lumasaaba results, the learner may still find a training-consistent hypothesis without selecting the generalization that best matches the adopted phonological analysis.

At the same time, systematic exploration of the (k, Φ) space reveals genuine structural boundaries. In certain grammatical systems such as vowel harmony, no configuration within the bounded k yields correct generalization, even if feature combinations were to be enriched. This suggests that the limitation is not merely parametric, but representational such that locality computed over the full surface string is sometimes insufficient to capture the relevant dependency.

The next chapter therefore introduces an additional representational layer, tiers, originally motivated in autosegmental phonology for tones (Goldsmith, 1976). By projecting only the elements relevant to a given process onto a restricted domain, tier-based representations preserve the commitment to low-k locality while allowing dependencies that are non-local on the surface string to be strictly local on the tier.

Chapter 7

Learning Tiers

The processes discussed and learned in previous chapters shared one important property of being *local* in the sense that they can be modeled and learned by giving the learner restricted access to memory bounded by a parameter k. However, cross-linguistic generalizations also include patterns whose basic generalization is difficult to express in terms of strict local adjacency, because the dependency requires large portions of the string to be effectively *ignored* (or absent) in order for the relevant elements to stand in a local relation. Canonical examples are harmony and dissimilation processes. In vowel harmony, a suffix vowel may agree in backness (and/or rounding) with a stem vowel across arbitrarily many intervening consonants, as in Turkish and other well-studied harmony systems (see e.g. Clements and Sezer 1982 for Turkish). Consonant harmony shows a parallel profile: in many systems, sibilants (or other consonant classes) agree in place-related features across intervening material, with the agreeing consonants separated by segments that neither undergo nor block the dependency (Rose and Walker, 2004). Long-distance dissimilation raises the same computational issue in the opposite direction, requiring elements to become *less* similar across distance (for instance, long-distance [labial] dissimilation is reported for Berber, and Tuareg exhibits long-distance labial dissimilation alongside sibilant harmony; see discussion in Alderete 2003 and the learning-motivated treatment in Burness and McMullin 2021).

In such *long-distance* patterns, the relationship between a trigger and a target may

hold across arbitrarily many intervening segments. For a learner that operates over surface windows of fixed size k, these unbounded dependencies are problematic such that no single k guarantees that both trigger and target will always be jointly visible. Moreover, long distance processes may co-occur with other dependencies of the same type in the same word, raising the possibility of computational interference when multiple predictors must be tracked in parallel (Burness and McMullin, 2021).

A standard representational response is to treat these dependencies as local on a *tier* that contains only the elements relevant to the process. This idea is closely related to the autosegmental treatment of tone, where tonal features are represented on a separate tier and linked to segmental structure rather than being forced into a single linear string (Goldsmith, 1976). Here, however, the representational motivation is not (only) to separate tonal melodies from segments, but to make explicit that many segmental dependencies are computed over a *restricted subsequence* of the string.

In computational terms, tier projection can be understood as an *erasure* mechanism. Given a tier alphabet $\Theta \subseteq \Sigma$, the projection π_Θ maps an input string $w \in \Sigma^*$ to a string $\pi_\Theta(w) \in \Theta^*$ by deleting all symbols not in Θ. Dependencies that are non-local on w can then be stated locally over the resulting tier string, i.e. over the elements $\theta \in \Theta$ rather than over the entire surface string (Heinz et al., 2011; Burness and McMullin, 2019; Burness et al., 2021).

From the perspective of computational analysis, many long-distance processes have been modeled in terms of locality imposed on their *outputs*. Classes of such mappings are referred to as *Output Strictly Tier-based Local* (OTSL) functions, and there are grammatical inference results showing that these mappings can be learned from positive data (Burness and McMullin, 2019; Burness et al., 2021). Adapting these learners to the approach advocated in this dissertation, however, is not immediate and requires introducing several changes. First, I adopt the insight from algebraic characterizations of phonological mappings from Lambert and Heinz (2023) who show how output-oriented processes can be expressed in an input-oriented way, which provides the key

bridge needed here. Second, although Burness and McMullin (2019) provide an algorithm for learning the content of an OTSL_2 tier (i.e. under the restriction $k = 2$), that learner shares a limitation familiar from other grammatical inference algorithms, including SOSFIA, in which successful identification can depend on receiving large and highly specific data. In this chapter we therefore propose a new learner built in the same overall fashion as Burness & McMullin's, accompanied by practical relaxations of the tier-induction procedure aimed specifically at the sparse-data settings. We then show that once tier content is inferred (and with $k = 2$ fixed), the resulting mapping can be represented as a deterministic transducer whose behavior falls within the algebraic class of tier-based definite functions ($\mathbf{D}_T$) (Lambert, 2023). Crucially, this means that after tier inference we recover a representation in the same input-local format used throughout the dissertation. Once the shape of the transducer is fixed, SEAL can be adapted to infer the outputs.

This chapter is organized as follows. Section 7.1 motivates tier-based representations for long-distance dependencies and introduces Turkish backness and roundness harmony as a guiding case study, showing why strictly local surface windows are insufficient without tier projection. Section 7.2 then develops an algebraic analysis of Turkish backness harmony, following the output-to-input bridge of Lambert and Heinz (2023), and argues that the resulting mapping falls within the tier-based definite class $[\![\mathbf{D}]\!]_T$ once the appropriate tier is fixed. Section 7.3 presents the grammatical-inference background from Burness and McMullin (2019) on learning canonical tiers in the 2-OTSL setting, and Section 7.3.2 adapts that procedure to the input-oriented 2-ITSL case, introducing the tier learner, `CanTIERr-2`, that is designed for sparse data. Section 7.4 reports a textbook-style experiment on the Turkish dataset of Ellis et al. (2022), showing that the proposed algorithm can recover linguistically plausible tier contents for both backness and roundness harmony and that these tiers support construction of the canonical transducer skeleton to which SEAL can then be applied to infer transition outputs. Section 7.5 concludes by summarizing what tier inference buys

the FxF(SEAL) architecture, clarifying the intended conservatism of `CanTIERr-2`, and outlining limitations and extensions.

7.1 From surface locality as k-OTSL function

In Turkish, the vowels of certain suffixes alternate based on characteristics of the stem vowel. For example, the plural suffix alternates as -lar or -ler and the genitive suffix can surface as either of the following forms: -in, -yn, -un or -ɯn as shown in (24) (Odden, 2005, p. 228-9).

(24)

NOM.SG	GEN.SG	NOM.PL	GEN.PL	Gloss
ip	ip-in	ip-ler	ip-ler-in	'rope'
ev	ev-in	ev-ler	ev-ler-in	'house'
kɯz	kɯz-ɯn	kɯz-lar	kɯz-lar-ɯn	'girl'
sap	sap-ɯn	sap-lar	sap-lar-ɯn	'stalk'

As can be seen, suffix vowels agree in backness with the stem vowel. Crucially, however, the trigger and the target may be separated by an unbounded number of consonants. As a consequence, no fixed surface window size k can guarantee successful inference.

Additionally, if the vowel in the noun root is round and the following vowel is high, it will adapt the roundness quality of the preceding vowel. However, if the following vowel is non-high it will not assimilate. Consider examples in 25 adapted from Odden (2005, p. 229).

(25)

NOM.SG	GEN.SG	NOM.PL	GEN.PL	Gloss
jyz	jyz-yn	jyz-ler	jyz-ler-in	'face'
pul	pul-un	pul-lar	pul-lar-ɯn	'stamp'
son	son-un	son-lar	son-lar-ɯn	'end'
kœj	kœj-yn	kœj-ler	kœj-ler-in	'village'

The roundness harmony is also blocked by a non-high vowel, such that the following high vowel will not be rounded even if the root has a rounded vowel, as in pularɯn. One way to analyze those phenomena is to project the relevant segments

on tiers. Clements and Sezer (1982) represent backness and roundness as separate "P-segment" on their own tier and are associated to vowels within the phonological word by general association conventions. Backness spreads straightforwardly so that suffix vowels match the stem's backness, while rounding is restricted because nonhigh vowels are opaque on the roundness tier and therefore do not undergo (and can interrupt) roundness association.

The spreading of the [back] and [round] features could also be thought of as an iterative process, such that the value for those features are determined based on the intermediate representation of the preceding vowel. In that sense, those processes could be modeled with k Output Strictly Local (k-OSL) functions (Chandlee et al., 2015) which can be represented with k-OSL deterministic transducers. The structure of such machines, similarly to previously introduced k-ISL DFTs, is fixed, such that the choice of each new output symbol is determined by a bounded window of the most recently produced output symbols. Each state in a k-OSL DFT corresponds to the last k output symbols, and transitions both emit the next output segment and update the state to reflect the newest output context. Because the processes discussed here are triggered, affected, and sometimes blocked, by vowels only, those segments must be projected to successfully model the processes as k-OSL functions. In other words, backness and rounding harmony in Turkish can be represented with a *tier-based* 2-OSL function (2-OTSL). Figure 7.1 presents 2-OTSL DFT for backness harmony. We assume that the underlying vowels that undergo changes in the suffixes lack specification for backness, and will be represented as E for [-high] suffixes and and I for [+high] suffixes. The states are limited to two representative vowels: back a and front e.

Once the representation is restricted to an appropriate tier alphabet, the harmony-relevant contexts collapse to just two equivalence classes, and therefore can be implemented with two states. Under a full phonemic inventory, the learner would instead need a much larger state space to represent all the relevant elements. Having shown that Turkish backness harmony can be represented as a 2-OTSL function, the next

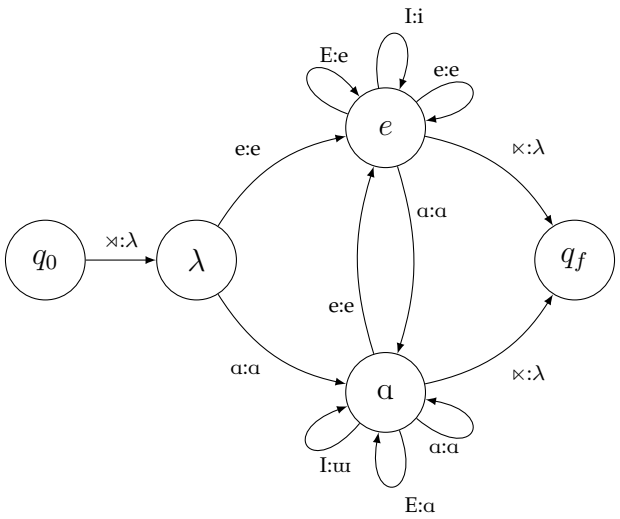


FIGURE 7.1: A 2-OTSL DFT modelling back harmony in Turkish for tier alphabets: $\Sigma = \{a, e, E, I\}$ and $\Delta = \{a, e, i, \text{ɯ}\}$.

section presents an algebraic characterization of this phonological mapping and concludes that Turkish harmony can be analyzed in terms of input as well.

7.2 Algebraic Analysis of backness harmony

This section summarizes the main idea behind the algebraic analysis of output-oriented processes, and shows how those algebraic properties allows of re-interpretation of those processes in terms of their inputs.

So far in this work phonological processes were modelled as string-to-string functions computed by deterministic subsequential transducers. These functions can be classified using subregular notions of locality (e.g. ISL, OSL, and their tier-based equivalents). An *algebraic* approach complements this picture by associating each subsequential function with a finite algebraic object, particularly its *syntactic semigroup*, and then using well-studied algebraic properties to diagnose the type of dependency the function implements (Lambert, 2022; Lambert and Heinz, 2023, 2024). This perspective is attractive for phonology because it unifies the classification of constraints and processes, and connects directly to known algorithmic techniques for classification and

learning in automata theory.

7.2.1 Preliminaries

A *tier* Θ is a subset of the alphabet Σ. The *tier projection* of a string w is the map $\pi_\Theta : \Sigma^* \to T^*$ defined recursively as follows: for the base case, $\pi_\Theta(\lambda) = \lambda$; for the inductive case, $\pi_\Theta(wa) = \pi_\Theta(w)a$ iff $a \in \Theta$ and $\pi_\Theta(wa) = \pi_\Theta(w)$ otherwise. Symbols that do not belong to the tier $(\Sigma \setminus \Theta)$ are called *neutral letters* and symbols in Θ are called *salient*.

A *semigroup* is a set S closed under an associative multiplication operation. An element $a \in S$ is *idempotent* whenever $aa = a$; if all elements of S are idempotent, then S is a *band*.

Lambert and Heinz (2024) define subsequential transducer as a 7-tuple $\tau = \langle Q, \Sigma, \Gamma, \delta, q_0, \rho, \sigma \rangle$, where Q is a finite set of states, Σ is a finite set of input symbols, Γ is a finite set of output symbols, $\delta : Q \times \Sigma \to Q \times \Gamma^*$ is the transition function, $q_0 \in Q$ is the initial state, $\rho \in \Gamma^*$ is a prefix prepended to all outputs, and $\sigma : Q \to \Gamma^*$ is a suffixing function. The machine processing function μ is defined recursively by the base case $\mu(q, \lambda, v) = v\,\sigma(q)$ and recurrence $\mu(q, au, v) = \mu(q', u, vw)$ where $a \in \Sigma$ and $\delta(q, a) = (q', w)$, and the function computed by τ is $f(w) = \mu(q_0, w, \rho)$. For every subsequential function f, there exists a *minimal onward* subsequential transducer that represents it,

Actions and semigroups and multiplication tables Given a set of the states in Q, $\langle q_0, \ldots, q_n \rangle$, the *action* given by a symbol $a \in \Sigma$ is the tuple $\langle q_0 \star a, \ldots, q_n \star a \rangle$. Distinct symbols may have the same action, in which case they exhibit the same behaviors. Importantly, neutral letters do not change state, which means that their action is always the identity action $\langle q_0, \ldots, q_n \rangle$, denoted 1. The actions given by individual letters form the basis of the *transition semigroup* denoted by S. If $x, y \in S$ are actions, then the product xy is the action obtained by applying x and then y. A convenient representation of S is its *multiplication table*, whose (x, y)-cell contains the product xy.

Multiplication tables directly expose which elements behave alike with respect to left- and right-composition, and hence make visible the key equivalence relations used in algebraic classification.

Green's relations and relevant complexity varieties. Many important algebraic classes of regular languages and subsequential functions can be defined using Green's relations (Green, 1951). Given a finite semigroup S, following (Colcombet, 2011) definitions preorders $\leq_L, \leq_R, \leq_J$ are expresses as follows:

$$a \mathrel{\mathcal{L}} b \text{ iff } a \in Sb \cup \{b\}, \qquad a \mathrel{\mathcal{R}} b \text{ iff } a \in bS \cup \{b\}, \qquad a \mathrel{\mathcal{J}} b \text{ iff } a \in SbS \cup Sb \cup bS \cup \{b\}.$$

The associated equivalence relations are L, R, J. A semigroup is $\mathcal{L}$-*trivial* if it has no two distinct $\mathcal{L}$-equivalent elements (and analogously for R-triviality). A semigroup is *definite* (variety $[\![\boldsymbol{D}]\!]$) iff it is $\mathcal{L}$-trivial and its only idempotents lie in the minimal $\mathcal{J}$-class; it is *reverse definite* (variety $[\![\boldsymbol{K}]\!]$) iff it is $\mathcal{R}$-trivial and its only idempotents lie in the minimal $\mathcal{J}$-class. These classes correspond, respectively, to dependencies on bounded *suffix* information and bounded *prefix* information. There is a crucial connection between k-ISL functions and $[\![\boldsymbol{D}]\!]$, such that they both describe the same class. This correspondence is especially important for this work, because once a phonological generalization lies with this variety, we can reasonably expect that learners designed for k-ISL functions, such as SEAL, will perform well on this type of pattern.

It is also possible to define tier-based equivalents of those classes. Tiers enter the algebraic picture through *neutral letters*, such that if a segment is irrelevant for the process, it may never change state, i.e. it induces the *identity* action on Q. The *tier-based definite* class is written as $[\![\boldsymbol{D}]\!]_T$ and *tier-based reverse definite* as $[\![\boldsymbol{K}]\!]_T$. For both of those classes all *non-identity* elements satisfy the relevant definiteness conditions.

7.2.2 Turkish back harmony as a tier-based definite process

Lambert and Heinz (2024) show while the perfect correspondence between k-ISL and algebraic varieties exist, the same is not true for the output-oriented corresponding class of k-OSL functions. Importantly, they also show that many such functions fall within the definite, reverse definite and their tier extensions classes, as long as they are analyzed via subsequential transducer in its minimal form. The algebra thus provides a complementary explanation of why these processes are learnable and how their apparent output-orientation can correspond to a very low-memory on the input after tier projection.

Recall the transducer in Figure 7.1 modeling Turkish back vowel harmony after projecting a and e on a tier. Following the notation adapted there, let the tier alphabets be $\Sigma = \{a, e, E, I, C\}$ and $\Delta = \{a, e, i, \text{ɯ}, C\}$, where C is any consonant, a and e are fully specified back/front vowels, and underspecified suffix vowels are represented as E ([-high]) and I ([+high]) and surface as [a,e] and [ɯ, i], respectively. The relevant memory is restricted to just the most recently produced vowel: after a front vowel $E \mapsto e$ and $I \mapsto i$, whereas after a back vowel $E \mapsto a$ and $I \mapsto \text{ɯ}$.

From the algebraic viewpoint, the computation is governed by a small set of state-update actions that encode the "harmonic setting" (front vs. back) and treat all non-tier segments as neutral. To make this explicit, consider the minimal subsequential transducer in Figure 7.2. It shows that the relevant control states are q_F and q_B, signifying that the last segment on the tier is Front and Back vowel, respectively.

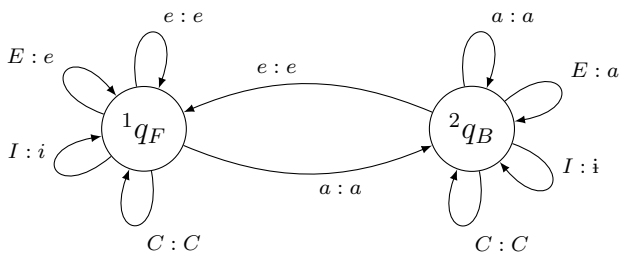


FIGURE 7.2: Minimal onward transducer for back harmony.

The 2-DFT in Figure 7.2 enables to define actions that arise from the input symbols from both states. The neutral segments C, I, E, self loop at each state, which can be represented as identity action $\mathbb{1} = \langle 1, 2 \rangle$, which means that when reading the segments from state 1 (q_F), the transducer goes back to state 1, and the same is true for state 2 (q_B). The remaining two actions can be understood as resetting the context to either back or front vowel: $f = \langle 1, 1 \rangle$ from input e and $b = \langle 2, 2 \rangle$ from input a. Those three actions form a semigroup $S = \{\mathbb{1}, f, b\}$, and their relations are presented in the multiplication table below.

$\cdot$	$\mathbb{1}$	f	b
$\mathbb{1}$	$\mathbb{1}$	f	b
f	f	f	b
b	b	f	b

This table encodes the key generalization: composing a front-setting and a back-setting action yields the action corresponding to the *most recently encountered* tier value (e.g. $fb = b$ and $bf = f$), while neutral segments contribute the identity action 1. Structurally, S is $\mathcal{L}$-trivial (distinct columns, hence distinct right-actions) but not $\mathcal{R}$-trivial (the rows for f and b contain the same elements, so $f \mathcal{R} b$), and every element is idempotent, i.e. $x^2 = x$ for all $x \in S$. The semigroup is nevertheless *not* definite because the identity 1 is an idempotent element that lies outside the minimal $\mathcal{J}$-class (in this example the minimal $\mathcal{J}$-class is $\{f, b\}$, since f and b generate the same smallest two-sided ideal and hence $f \mathcal{J} b$). However, once the neutral identity action is set aside, the remaining *non-identity* elements still satisfy the definiteness conditions, i.e., all actions are idempotent among the non-identity elements occur in the minimal $\mathcal{J}$-class—so S falls in the tier-based definite class $[\![\mathbf{D}]\!]_T$ in the sense of Lambert (2023).

What we have just shown is that the output-oriented 2-OTSL analysis of Turkish backness harmony falls within an equivalent algebraic characterization stated in input-oriented terms. In order for the learner developed in this dissertation (SEAL) to succeed on such a pattern, we must also determine the *content of the tier*, since tier

specification is required to construct the relevant minimal onward deterministic transducer in the input-oriented setting. The next section therefore summarizes theoretical results from grammatical inference on learning tier content for the 2-OTSL class. We then adapt the procedure to the input-oriented case and propose a relaxation of its conditions to allow successful inference in small-data settings.

7.3 Learning the Tier

Burness and McMullin (2019) propose a structured learner for inferring the content of the tier for 2-OTSL functions. They do so by assuming that there exists a canonical 2-tier, a tier for which the memory window $k = 2$. Since they do so specifically for *output*-oriented processes, the tier is first assumed to be equal to the output alphabet Δ, and next elements from the tier are erased (in the sense of Jardine and Heinz (2016)) if they do not meet certain formal requirements resulting from the properties of OTSL functions. Before we introduce a complementary method to infer the elements on the tier for 2-ITSL functions, we provide some technical background.

7.3.1 Preliminaries

First, we adapt a prefix function (Definition 7.1) defined in Chandlee et al. (2015) which determines what can be *surely* output reading any input prefix. This function also separates outputs when reading elements $x \in \Sigma$ from the ones produced in the final transitions (transition to the final state, here reading the right boundary marker $\rtimes$).

Definition 7.1. *Prefix function f^p is a function associated with a subsequential function f such that $f^p(w) = \texttt{lcp}(f(w\Sigma^*))$.*

With the prefix function it is now possible to calculate contribution attributed to a single element from Σ. The contribution function `cont` is defined in Definition 7.2 for

the two cases when a prefix u is extended by an element $\sigma \in \Sigma$ and the right boundary element marking end of the string ($\ltimes$).

Definition 7.2. *For a subsequential function f, element $\sigma \in \Sigma$, $\ltimes \notin \Sigma$, and some string $x \in \Sigma^*$:*

$\texttt{cont}_f(u, \sigma) = f^p(u)^{-1} \cdot f^p(u\sigma)$, *and*

$\texttt{cont}_f(u, \ltimes) = f^p(u)^{-1} \cdot f(u)$.

In the same way the class of k-ISL functions was defined in Definition 2.3 in terms of suffix and tails equivalency, a k-OSL will be defined. The only difference is that it is not the input suffix but output suffix that determines the equivalency.

Definition 7.3. *A function f is OSL iff there is a $k \in \mathbb{N}$ such that for all $u_1, u_2 \in \Sigma^*$ if $Suff^{k-1}(f^p(u_1)) = Suff^{k-1}(f^p(u_2))$ then $tails_f(u_1) = tails_f(u_2)$. A function is k-OSL if it is OSL for some k.*

A tier is simply a subset of alphabet symbols. So the elements of the tier can be acquired by removing certain elements from from a tier that are present in the alphabet. This is achieved by the erasure function (adapted from Burness and McMullin (2019)).

Definition 7.4. *Given an alphabet Σ, a tier $\Theta \subseteq \Sigma$, and a string $w = a_1 ... a_n$, $\texttt{Erase}_\Theta(w) = b_1 ... b_n$ where $\forall i \leq n, b_i = a_i$ if $a_i \in \Theta$, else $b_i = \lambda$.*

Given a tier Θ which is a subset of elements from Σ, Burness and McMullin (2019) define k-Output Tier-based Strictly Local Functions (k-OTSL; Definition 7.5) as a class that properly contains the OSL function class. Analogically, a class of k-Input Tier-based Strictly Local Functions (k-ITSL) is provided in Definition 7.7 (Burness and McMullin, 2020).

Definition 7.5. *A function f is k-OTSL iff there is a $k \in \mathbb{N}$ and a tier $\Theta \subseteq \Delta$ such that for all $u_1, u_2 \in \Sigma^*$ if $Suff^{k-1}_\Theta(f^p(u_1)) = Suff^{k-1}_\Theta(f^p(u_2))$ then $tails_f(u_1) = tails_f(u_2)$.*

Definition 7.6. *A function f is k-ITSL iff there is a $k \in \mathbb{N}$ and a tier $\Theta \subseteq \Sigma$ such that for all $u_1, u_2 \in \Sigma^*$ if $Suff^{k-1}_\Theta(u_1) = Suff^{k-1}_\Theta(u_2)$ then $tails_f(u_1) = tails_f(u_2)$.*

In their later work (Burness and McMullin, 2021), the definition was extended to multiple tiers to account for such processes like height and nasal harmony in Kikongo that require projection of two disjoint tiers. This work is limited to learning one tier only.

Burness and McMullin (2019) also prove that for each 2-OTSL function there exists a unique *canonical 2-tier* Θ which is a superset of any 2-tier for f. In other words, a canonical 2-tier is the largest tier. They then build on this another important property that is specifically used to learn the 2-tier, which we adapt and present below.

In particular, their Lemma 5 shows that any strict superset of the canonical tier is too large. Concretely, if $\Theta \subset \Omega \subseteq \Delta$, then there must exist an "extra" symbol $a \in (\Omega \backslash \Theta)$ and two input prefixes $u_1, u_2 \in \Sigma^*$ such that the most recent Ω-tier symbol in the prefix outputs is the same, i.e. $\texttt{Suff}^1_\Omega(f_p(u_1)) = \texttt{Suff}^1_\Omega(f_p(u_2)) = a$, but there is some continuation symbol $x \in \Sigma \cup \{n\}$ for which the function behaves differently in the two contexts, witnessed by a mismatch in contributions ($\texttt{cont}_f(x, w_1) \neq \texttt{cont}_f(x, w_2)$). Intuitively, if we "force" a symbol onto the tier that does not truly belong to it, then there will be contexts in which treating that symbol as the most recent tier symbol incorrectly predicts what the mapping will contribute next.

In order to show analogous procedure will work for 2-ITSL function, we must show that if there exist more than one 2-tier to describe such function, the union of any such 2-tier is also a 2-tier for that function.

Proposition 2. *If f in an 2-ITSL function for which there are two 2-tier $\Theta_1 \subseteq \Sigma$ and $\Theta_2 \subseteq \Sigma$, then $\Omega = \Theta_1 \cup \Theta_2$ is also a 2-tier for f.*

Proof. Assume $\texttt{Suff}^1_\Omega(w_1) = \texttt{Suff}^1_\Omega(w_2) = t$. Then $t \in \Omega$, hence $t \in \Theta_1$ or $t \in \Theta_2$. If $t \in \Theta_1$, then $\texttt{Suff}^1_{\Theta_1}(w_1) = \texttt{Suff}^1_{\Theta_1}(w_2) = t$, and since Θ_1 is a 2-tier for f we get $\texttt{tails}_f(w_1) = \texttt{tails}_f(w_2)$. If $t \in \Theta_2$, then $\texttt{Suff}^1_{\Theta_2}(w_1) = \texttt{Suff}^1_{\Theta_2}(w_2) = t$, and since Θ_2 is a 2-tier for f we again get $\texttt{tails}_f(w_1) = \texttt{tails}_f(w_2)$. Therefore $\texttt{Suff}^1_\Omega(w_1) = \texttt{Suff}^1_\Omega(w_2)$ implies $\texttt{tails}_f(w_1) = \texttt{tails}_f(w_2)$, so Ω is a 2-tier for f. □

Having shown the union closure allows for identifying a canonical 2-tier for any 2-ITSL function in the same way Burness and McMullin (2019) did. It is important to emphasize that this property hold only for $k = 2$ because with this memory window, the suffixes under consideration are exactly of length 1. That leads to a formal definition of a canonical 2-tier for 2-ITSL functions.

Definition 7.7. *A tier Θ is a canonical 2-tier for a 2-ITSL function f iff there is no other tier $\Omega \subseteq \Sigma$ for f s.t. $|\Omega| \geq |\Theta|$.*

Finally, now plugging the properties of 2-ITSL function in Burness and McMullin (2019)'s Lemma 5 gives us a property that can be directly used to learn the content of canonical 2-tier. Proposition 3 shows that if two strings have the same input suffix, it implies that the same contribution to those two strings should also be the same.

Proposition 3. *Let $f : \Sigma^* \to \Delta^*$ be a 2-ITSL function, let $\Theta \subseteq \Sigma$ be its canonical 2-tier, and let Ω be such that $\Theta \subset \Omega \subseteq \Sigma$. Then there exist $a \in (\Omega \setminus \Theta)$, $w_1, w_2 \in \Sigma^*$, and $x \in \Sigma \cup \{\ltimes\}$ such that $\mathtt{Suff}^1_\Omega(w_1) = \mathtt{Suff}^1_\Omega(w_2) = a$ and $\mathtt{cont}_f(w_1, x) \neq \mathtt{cont}_f(w_2, x)$.*

Proof of this proposition follows the exact same logic as the proof for Lemma 5 in Burness and McMullin (2019) with the only difference being that the suffix is taken from the input and not the output so all $\mathtt{Suff}^1_\Omega(f^p(w_i))$ becomes $\mathtt{Suff}^1_\Omega(w_i)$.

7.3.2 Learning 2-tier for 2-ITSL functions

The learning of the tier content starts with the assumption that all the elements of Σ are present on the tier. Next, leveraging the properties of Proposition 3 and consistently calculating which elements do not follow this property and erase them. The calculation proceeds with two major steps: build a set of tuples of all valid prefixes p and the `lcp` of the outputs mapped from input strings starting with p. A valid p is one for which there exists $\sigma \in \Sigma$ for which $p\sigma$ is also a prefix. The procedure is called `PFE` and its pseudo code is provided below.

Algorithm 7: Prefix Function Estimation (PFE)

Data: A sample $S \subseteq \Sigma^* \times \Delta^*$
Result: A set $P \subseteq \Sigma^* \times \Delta^*$ estimating the prefix function f_p

$P \leftarrow \emptyset$
$X \leftarrow \{\, x \mid x \in \texttt{Pref}^*(w),\ \text{where } (w, u) \in S \,\}$
$Y \leftarrow \{\, x \in X \mid (\forall a \in \Sigma)\, [xa \in X] \,\}$
for all $y \in Y$ **do**
  $z \leftarrow \texttt{lcp}(\{\, u \mid (w, u) \in S,\ \text{where } y \in \texttt{Pref}^*(w) \,\})$
  $P \leftarrow P \cup \{(y, z)\}$
end for
return P

Given the set P which is what PFE returns, the learner loops through the elements $\theta \in \Theta$, with the starting point being $\Theta = \Sigma$. In order to determine whether θ should be deleted from the tier, the learner selects elements $(y, z) \in P$ for which $\texttt{Suff}^1_\Theta(y) = \sigma$ and creates a temporary set Match. Next, given this set, the learner iterates through elements $\sigma \in \Sigma$ and checks whether $\texttt{cont}(y, \sigma)$ is consistent for all the prefixes y in Match. If there is more than one contribution for the suffix y in Match, θ gets added to a set Remove. Finally, the elements of Remove are subtracted from Θ and the tier is returned. The pseudo code for this procedure is presented in Algorithm 8.

Algorithm 8: 2-ITSL$_2$ Canonical Tier Inference (CanTIER-2)

Data: Sample $S \subseteq \{\rtimes\}\Sigma^*\{\ltimes\} \times \Delta^*$
Result: An inferred 2-tier $\Theta \subseteq \Sigma$

$P \leftarrow \texttt{PFE}(S)$
$\Theta \leftarrow \Sigma$
$\texttt{Remove} \leftarrow \{*\}$
while $\texttt{Remove} \neq \emptyset$ **do**
  $\texttt{Remove} \leftarrow \emptyset$
  for all $\theta \in \Theta$ **do**
    $\texttt{Match} \leftarrow \{(y, z) \in P \mid \texttt{Suff}^1_\Theta(y) = \theta\}$
    for all $\sigma \in \Sigma$ **do**
      $C_{\theta,\sigma} \leftarrow \{\, z^{-1} \cdot x \mid (y, z) \in \texttt{Match} \wedge (y\sigma, x) \in P \,\}$
      if $|(C_{\theta,\sigma})| > 1$ **then**
        $\texttt{Remove} \leftarrow \texttt{Remove} \cup \{\theta\}$
        break
      end if
    end for
    if $\theta \notin \texttt{Remove}$ **then**
      $C_{\rtimes,\theta} \leftarrow \{\, z^{-1} \cdot u \mid (y, z) \in \texttt{Match} \wedge (y, u) \in S \,\}$
      if $|(C_{\rtimes,\theta})| > 1$ **then**
        $\texttt{Remove} \leftarrow \texttt{Remove} \cup \{\theta\}$
      end if
    end if
  end for
  $\Theta \leftarrow \Theta \setminus \texttt{Remove}$
end while
return Θ

7.3.3 Example

Table 7.1 lists the featural specifications for the seven segments distinguished in this analysis. The examples in (7.3.3) are hand-crafted from these representations by selecting the relevant projected featural samples.

	t͡ʃ	**k**	**i**	ɯ	**E**	**I**	**e**	ɑ
cons	+	+	−	−	−	−	−	−
high	0	+	+	+	−	+	−	−
front	0	0	+	−	0	0	+	−
back	0	0	−	+	0	0	−	+
tense	0	0	+	+	+	+	+	0
low	0	−	−	−	−	−	−	+

TABLE 7.1: A subset of Turkish phonemic inventory used to show tier learning for back harmony.

(26)

$\mathbf{S}$	$\boldsymbol{S}_{([back], back)}$
ik-I$\widehat{\text{tʃ}}$ ↦ ik$\widehat{\text{itʃ}}$	$-\ 0\ 0\ 0 \mapsto -\ 0\ -\ 0$
ɯ$\widehat{\text{tʃ}}$-Ek ↦ ɯ$\widehat{\text{tʃ}}$ak	$+\ 0\ 0\ 0 \mapsto +\ 0\ +\ 0$
ik-Ek-I$\widehat{\text{tʃ}}$ ↦ ik$\widehat{\text{ekitʃ}}$	$-\ 0\ 0\ 0\ 0\ 0 \mapsto -\ 0\ -\ 0\ -\ 0$
ɑk-Ek-I$\widehat{\text{tʃ}}$ ↦ ik$\widehat{\text{ɑkɯtʃ}}$	$+\ 0\ 0\ 0\ 0\ 0 \mapsto +\ 0\ +\ 0\ +\ 0$
ɑkik-Ek-I$\widehat{\text{tʃ}}$ ↦ ɑk$\widehat{\text{ikekitʃ}}$	$+\ 0\ -\ 0\ 0\ 0\ 0\ 0 \mapsto +\ 0\ -\ 0\ -\ 0\ -\ 0$
et$\widehat{\text{ʃɯk}}$-Ek-I$\widehat{\text{tʃ}}$ ↦ et$\widehat{\text{ʃɯkɑkɯtʃ}}$	$-\ 0\ +\ 0\ 0\ 0\ 0\ 0 \mapsto -\ 0\ +\ 0\ +\ 0\ +\ 0$
ɑkɯk-Ek-I$\widehat{\text{tʃ}}$ ↦ ɑkɯk$\widehat{\text{ɑkɯtʃ}}$	$+\ 0\ +\ 0\ 0\ 0\ 0\ 0 \mapsto +\ 0\ +\ 0\ +\ 0\ +\ 0$

With the projected $\boldsymbol{S}_{([back], back)}$ sample, the SEAL learner will fail to predict desired output because it cannot differentiate the 0 value for feature [back] for consonants vs. the underspecified vowel (marked in red on the input side and in green on the output for better readability). However, if the input gets additional information from feature [cons], the projected sample correctly distinguishes the target. Data below shows the projected $\boldsymbol{S}_{(\left[\begin{smallmatrix}\text{back}\\ \text{cons}\end{smallmatrix}\right], back)}$ sample.

(27)

$$\boldsymbol{S}_{(\left[\begin{smallmatrix}\text{back}\\ \text{cons}\end{smallmatrix}\right], back)}$$

$$\begin{bmatrix} -&0&0&0\\ -&+&-&+ \end{bmatrix} \mapsto -\ 0\ -\ 0$$

$$\begin{bmatrix} +&0&0&0\\ -&+&-&+ \end{bmatrix} \mapsto +\ 0\ +\ 0$$

$$\begin{bmatrix} -&0&0&0&0&0\\ -&+&-&+&-&+ \end{bmatrix} \mapsto -\ 0\ -\ 0\ -\ 0$$

$$\begin{bmatrix} +&0&0&0&0&0\\ -&+&-&+&-&+ \end{bmatrix} \mapsto +\ 0\ +\ 0\ +\ 0$$

$$\begin{bmatrix} +&0&-&0&0&0&0\\ -&+&-&+&-&+&- \end{bmatrix} \mapsto +\ 0\ -\ 0\ -\ 0$$

$$\begin{bmatrix} -&0&+&0&0&0&0\\ -&+&-&+&-&+&- \end{bmatrix} \mapsto -\ 0\ +\ 0\ +\ 0$$

$$\begin{bmatrix} +&0&+&0&0&0&0\\ -&+&-&+&-&+&- \end{bmatrix} \mapsto +\ 0\ +\ 0\ +\ 0$$

The projected sample in (27) now does provide sufficient information to distinguish the elements that undergo the change from those that do not. The input alphabet inferred from this sample is $\Sigma_{(\Phi,\phi)} = \{[\begin{smallmatrix}-\\ -\end{smallmatrix}], [\begin{smallmatrix}0\\ +\end{smallmatrix}], [\begin{smallmatrix}0\\ -\end{smallmatrix}], [\begin{smallmatrix}+\\ -\end{smallmatrix}]\}$.

The tier-learning procedure begins, as described in the previous section, by constructing a set of valid prefixes and their 1-suffixes. Next, iterating over the input alphabet $\sigma \in \Sigma$, `CanTIER-2` checks whether appending σ to every prefix ending in $a \in \Theta$ yields the same contribution.

However, the toy dataset is very limited and therefore would not be sufficient to learn the tier content. In fact, the characteristic sample required by `CanTIER-2` would

be difficult to obtain from real-world data, assuming it can be obtained at all. For Turkish, for example, underspecified elements are not expected to appear in roots, which already suggests that the characteristic sample needed by `CanTIER-2` may never be attested for Turkish backness harmony unless more structural information is provided, such as morpheme boundaries. More generally, some languages impose strict syllable-structure constraints (e.g., banning consecutive vowels or consecutive consonants), which would likewise make learning 2-tier dependencies impossible with the current version of the learner.

We therefore propose relaxed versions of `PFE` and `CanTIER-2`, called `PFEr` and `CanTIERr-2`, r for relaxed, that are better suited for learning from small datasets by loosening some of the restrictions. Although these relaxed variants do not admit provable completeness guarantees, they rely on essentially the same underlying mechanism. The new algorithms are presented in Algorithm 9 and Algorithm 10.

Algorithm 9: Prefix Function Estimation-relaxed (`PFEr`)

Data: A sample $S \subseteq \Sigma^* \times \Delta^*$
Result: A set $P \subseteq \Sigma^* \times \Delta^*$ estimating the prefix function f_p

$P \leftarrow \emptyset$
$X \leftarrow \{\, x \mid x \in \texttt{Pref}^*(w),$ where $(w, u) \in S \,\}$
$Y \leftarrow \{\, x \in X \mid (\exists a \in \Sigma)\, [xa \in X] \,\}$ ▷ r-change
for all $y \in Y$ **do**
  $z \leftarrow \texttt{lcp}(\{\, u \mid (w, u) \in S,$ where $y \in \texttt{Pref}^*(w) \,\})$
  $z \leftarrow z[0 : |y|]$ ▷ cap `lcp` length (no lookahead)
  $P \leftarrow P \cup \{(y, z)\}$
end for
return P

Algorithm 10: 2-ITSL Canonical Tier Inference-relaxed (CanTIERr-2)

Data: Sample $S \subseteq \{\rtimes\}\Sigma^*\{\ltimes\} \times \Delta^*$
Result: An inferred 2-tier $\Theta \subseteq \Sigma$

$P \leftarrow \texttt{PFE-r}(S)$
$\Theta \leftarrow \Sigma \setminus \{\ltimes\}$
$\texttt{Remove} \leftarrow \{*\}$
while $\texttt{Remove} \neq \emptyset$ **do**
  $\texttt{Remove} \leftarrow \emptyset$
  for all $\theta \in \Theta$ **do**
    $\texttt{Match} \leftarrow \{(y, z) \in P \mid \texttt{Suff}^1_\Theta(y) = \theta\}$
    $\texttt{Next}(\theta) \leftarrow \{\, \sigma \in \Sigma \mid \exists (y, z) \in \texttt{Match}\ \exists x\, [(y\sigma, x) \in P] \,\}$ ▷ r-change
    for all $\sigma \in \Sigma$ such that $\exists (y, z) \in \texttt{Match}\ \exists x\, ((y\sigma, x) \in P)$ **do**
      $W_{\theta,\sigma} \leftarrow \{\, (y,\ \texttt{first}(z^{-1}{\cdot}x)) \mid (y, z) \in \texttt{Match} \wedge (y\sigma, x) \in P \,\}$ ▷ r-change
      if $\exists (y_1, r_1), (y_2, r_2) \in W_{\theta,\sigma}$ such that $y_1 \neq y_2 \wedge r_1 \neq r_2$ **then**
        $\texttt{Remove} \leftarrow \texttt{Remove} \cup \{\theta\}$
        break
      end if
    end for
    if $\theta \notin \texttt{Remove}$ **then**
      $W_{\ltimes,\theta} \leftarrow \{\, (y,\ \texttt{first}(z^{-1}{\cdot}u)) \mid (y, z) \in \texttt{Match} \wedge (y, u) \in S \,\}$ ▷ r-change
      if $\exists (y_1, r_1), (y_2, r_2) \in W_{\ltimes,\theta}$ such that $y_1 \neq y_2 \wedge r_1 \neq r_2$ **then**
        $\texttt{Remove} \leftarrow \texttt{Remove} \cup \{\theta\}$
      end if
    end if
  end for
  $\Theta \leftarrow \Theta \setminus \texttt{Remove}$
end while
return Θ

The original formulation of prefix-function estimation and tier inference assumes a strong form of prefix closure: a prefix y is only considered if *every* one-symbol extension ya (for all $a \in \Sigma$) is also observed as a prefix in the sample. While this condition supports clean learnability arguments under characteristic samples, it is often too restrictive for realistic (small) datasets, where most prefixes have only a few attested continuations. While featural representation of data might help overcome this issue to some extent, it will most likely not eliminate the problem entirely. We therefore replace the closure requirement by the weaker notion of *internal prefixes*, namely those prefixes y for which at least one extension ya is observed in the sample (as opposed $\forall a \in \Sigma$). This modification ensures that PFEr returns a nontrivial set P even when the sample

contains only a limited subset of the full paradigm. In other words, this extension still allows to collect valid, for the given sample, information about f^p.

Because the learner no longer assumes full extension closure, `CanTIERr-2` also avoids quantifying over *all* $\sigma \in \Sigma$ when checking consistency conditions. Instead, it tests only *attested continuations* σ for which some pair $(y\sigma, x) \in P$ exists with $(y, z) \in P$ and $\texttt{Suff}^1_\Theta(y) = \theta$. This way the learner adequately adjusts to the data and extracts the most information possible from what is available.

A second obstacle in sparse samples comes from how `PFEr` estimates the prefix function $f^p(w)$: it sets $f^p(w)$ to the `lcp` of the outputs of all sample strings whose inputs share prefix w. When the sample provides only a single continuation for a given w, the resulting `lcp` collapses to the entire observed output and the accidental lack of variation as evidence about what must be produced "as early as possible" is treated as evidence. In tier learning, this can create spurious mismatches in estimated contributions and, in turn, spuriously license tier deletions.

To keep tier elimination conservative, `CanTIERr-2` therefore adopts a *double-witness* requirement, where a candidate symbol θ is removed from the current tier hypothesis only if the learner finds two distinct contexts, i.e. two distinct prefixes $y_1 \neq y_2$, for which appending the same σ yields different estimated contributions. Intuitively, the learner must see the contradiction twice, in two independent environments, before making the irreversible update of deleting θ from Θ. This preserves the intended learnability profile such that when the data are sufficiently informative, contradictions recur and the learner converges. However, when the data is sparse, the learner is allowed to fail to converge and instead return a conservative *superset* tier.

Finally, introducing the *seen twice* condition is beneficial in the sense that it will not eliminate a candidate tier symbol in cases where there exists only one relevant prefix that can be extended by a particular σ. For example, suppose there were an additional data point like this: $\left[\begin{smallmatrix} + & 0 & + & 0 & + & 0 & 0 & 0 & 0 \\ - & + & - & + & + & - & + & - & + \end{smallmatrix}\right] \mapsto \; + \, 0 + 0 + + \, 0 + 0$. Assume this is the only example in the sample that contains a prefix for which $\texttt{Suff}^1 = \left[\begin{smallmatrix} + \\ + \end{smallmatrix}\right]$, which for

example corresponds to a consonant [q]. Then, based on this single environment, we would not want the learner to make strong predictions and decide to delete $[\begin{smallmatrix}+\\+\end{smallmatrix}]$ from the tier.

Nevertheless, this restriction does not prevent cases in which there are, for example, two strings starting with some prefix w_1 for which there is exactly one distinct continuation through σ. In such cases, computing $\texttt{lcp}(w_1)$ can yield an output that is longer than expected (i.e., it includes material that should be attributed to the extension), and it can leave the inferred contribution of σ in $w_2 = w_1\sigma$ empty. To make this concrete, assume that in addition to the data in (27) there is an example as follows: $\left[\begin{smallmatrix}+&0&+&0&+&0&0&0&0\\-&+&-&+&-&-&+&-&+\end{smallmatrix}\right] \mapsto \texttt{+0+0++0+0}$. Now, when it is compared with the last example from the dataset (repeated here for clarity), $\left[\begin{smallmatrix}+&0&+&0&0&0&0&0\\-&+&-&+&-&+&-&+\end{smallmatrix}\right] \mapsto \texttt{+0+0+0+0}$. we can observe that they both begin with $w_1 = \left[\begin{smallmatrix}+&0&+\\-&+&-\end{smallmatrix}\right]$. Now during tier learning process there will be a step where $\theta = [\begin{smallmatrix}+\\-\end{smallmatrix}]$ is tested for deletion with $\sigma = [\begin{smallmatrix}0\\+\end{smallmatrix}]$. First, $\texttt{lcp}(w_1)$ outputs $\texttt{+0+0}$ because the only continuation of that string in the sample is via that σ so output material that should be *actually* attributed to σ is produced too early, already at w_1. Explicitly, $\texttt{lcp}(w_2) = \texttt{+0+0}$, so the contribution of σ in w_1 will then be equivalent to an empty string. Consequently, the learner will be exposed to inconsistent contributions of $[\begin{smallmatrix}0\\+\end{smallmatrix}]$ as other prefixes ending with $[\begin{smallmatrix}+\\-\end{smallmatrix}]$ that exhibit extensions with more than one element of Σ may assign different contributions. This will result in incorrectly deleting this θ from the tier.

One way to fix this would be to further restrict which prefixes are considered "usable": for instance, if a prefix w appears in the dataset but its only extension is with some $\sigma \in \Sigma$, then we would not consider w. We could go even further and restrict $w\sigma$ such that if it also has only one extension, then it would be dropped from consideration. After some experiments, we concluded that doing so restricts the candidates too much and creates situations where none of the Θ elements are ever deleted due to lack of "usable" evidence.

There has to be, then, another way to prevent $\texttt{lcp}$ from *overcommitting* to output

material that is only supported by a single continuation. In what follows, we therefore adopt an additional restriction that is motivated by standard properties of tier-based long-distance segmental dependencies. Long-distance segmental processes on tiers (excluding tonal cases such as opposite tonal insertion in Shupamem; Markowska et al., 2021) typically do not involve segment deletion or insertion but rather exhibit assimilatory processes. Moreover, this chapter restricts tier learning to the class of 2-ITSL processes, which means that in the direction of processing where the trigger is seen before the target, the target will be adjusted accordingly without any "waiting (λ) transitions". Therefore under the assumption that no deletion or insertion is expected to occur on the tier, it is reasonable to require that $\texttt{lcp}(w)$ not "run ahead" of w; specifically, we assume that $|\texttt{lcp}(w)| = |w|$. We emphasize that the right direction of processing is essential for this to work.

Introducing this restriction, which is compatible with SEAL's first learning step, effectively imposes a 1-to-1 alignment at the level of prefix estimation. Why does it solve the problem? Because now if $|w_1| = 3$, then $|\texttt{lcp}(w_1)| = 3$, which limits the output to $+\,0\,+$.

Then, once the next prefix is extended by σ, i.e. $w_2 = p_1\sigma$, the algorithm computes the correct contribution of σ, which in turn matches the contributions obtained for other prefixes in `Match` for the same $\theta \in \Theta$. Crucially, the restriction enforces $|\texttt{lcp}(p)| = |p|$, preventing $\texttt{lcp}(p)$ from incorporating output material licensed only after reading σ. The resulting incremental output attributed to the extension by σ is therefore evaluated as a single-symbol contribution (via `first`), and these one-symbol contributions are the objects compared across contexts in the nested loops of `CanTIERr-2`.

The lines in the pseudocode reflecting the main changes are marked with the *r-change* comment, where r stands for *relaxed*.

Once the tier content has been inferred, the corresponding memory bound k is fixed as well. In the present setting the tier window is always $k = 2$, since `CanTIERr-2`

is restricted to 2-ITSL functions. Following Lambert and Heinz (2024), we can then construct the canonical subsequential transducer associated with this tier and window. The tier elements provide the contextual distinctions that must be represented by separate states. Symbols that are off the tier do not update the tier context and are therefore implemented as self-loops on each state.

With the transducer *structure* fixed, the remaining task is to determine the output labels on transitions. To do so, we use SEAL in the same way as in the previous chapters: first, output variables O_i are assigned to transitions along the sample-consistent paths determined by the canonical transducer, and then their concrete values are inferred by one of the SEAL's modes. The next section provides a worked example of this procedure and reports results for tier learning on textbook-style Turkish datasets, covering both back and round harmony.

7.4 Example

This section illustrates the empirical behavior of `CanTIERr-2` on a textbook-style Turkish dataset exhibiting both backness and roundness harmony. We use the Turkish dataset reported in Ellis et al. (2022). The goal here is intentionally narrow: we ask whether the relaxed learner can recover linguistically plausible *tier content* from realistic sample sizes, rather than optimizing the end-to-end FxF(SEAL) pipeline. For that reason, we provide the relevant featural inventory Φ upfront. In later experiments, tier learning must be integrated into the full FxF(SEAL) system. We return to this point in the Discussion section.

The dataset contains 265 input–output pairs. For the purposes of this section, we removed items exhibiting vowel hiatus resolution through j-epenthesis, since the present chapter focuses on tier-based assimilatory mappings and the relaxed assumptions introduced above are not designed to handle insertion/deletion on tier. As in earlier sections, we also collapse vowel length, since length does not affect either of the

harmony processes considered here and would require introducing additional feature to the system. The full phonemic inventory in the dataset is provided in Appendix C. In this section we restrict attention to the subset of features chosen for the experiment, both for clarity and because these features are sufficient to state the targeted dependencies.

	j	o	u	i	ɯ	y	e	ɑ	I	E	b	k
back	−	+	+	−	+	−	−	+	0	0	0	0
front	+	−	−	+	−	+	+	−	0	0	0	0
round	−	+	+	−	−	+	−	−	0	−	−	−
low	−	−	−	−	−	−	−	+	−	0	0	−
tap	−	0	0	0	0	0	0	0	0	0	−	−
high	+	−	+	+	+	+	−	−	+	−	0	+

TABLE 7.2: Turkish phonemic inventory: all vowels and two representative consonants .

Table 7.2 summarizes the featural encoding used in this experiment. Two vowels are underspecified: I and E. The former is underspecified for [BACK], [FRONT], and [ROUND]; the latter is underspecified for [BACK], [FRONT], and [LOW], reflecting that it does not participate in roundness harmony. The additional underspecification of [FRONT] and [LOW] for E is required because these values vary across its surface allophones. Feature [HIGH] is necessary to distinguish the two harmony types (roundness harmony being restricted to high vowels in the relevant environments), and feature [TAP] serves as a coarse separator between vowels and consonants. While [TAP] is not a standard descriptive choice for this class of phenomena, it is used here as a practical representational device: it separates the approximant /j/ from the vowel classes in this feature system. In preliminary experiments, [CONSONANTAL] did not reliably distinguish /i/ from /j/ in this encoding, which led to incorrect generalizations and, in turn, inappropriate tier deletions. We view this as a representation issue rather than a limitation of the tier learner per se as the learner can only exploit distinctions made available by Φ, given the used feature system.

The tier-based processes we aim to learn can be stated as follows:

(28) Backness Harmony

$$\text{E} \longrightarrow / \left[\alpha \text{ back} \right] \_\_$$

(29) Backness and Roundness Harmony

$$\text{I} \longrightarrow \left[\begin{smallmatrix} \alpha \text{ back} \\ \alpha \text{ round} \end{smallmatrix} \right] / \left[\begin{smallmatrix} \alpha \text{ back} \\ \alpha \text{ round} \end{smallmatrix} \right] \_\_$$

7.4.1 Results

Table 7.3 reports (i) which feature sets Φ were used to learn tiers for each target feature ϕ (in bold), (ii) the resulting tier contents Θ, and (iii) the corresponding segmental interpretations.

At first glance, some of the learned tier contents may appear unexpected. In particular, the inferred tiers are not always *minimal* in the sense of containing only the segments that one might posit under an idealized analysis (e.g. only fully specified back vs. front vowels for backness harmony). Nevertheless, given the feature encoding adopted here and the limited range of contexts attested in the dataset, the outcomes are entirely satisfactory: `CanTIERr-2` deletes only those feature bundles for which the data provide *double-witness* evidence of inconsistency, and it otherwise retains candidates conservatively. The resulting tiers are therefore best understood as linguistically plausible *supersets* of a minimal tier hypothesis that are linguistically sound with respect to the observed evidence, and appropriately cautious when the sample does not yet support further eliminations.

Back and front For target [BACK] and [FRONT], the successful feature sets partition the inventory into back vowels, front vowels, underspecified vowels, the approximant /j/, and consonants. In both settings, the consonantal bundle is successfully deleted from the tier. What may seem surprising is that the approximant and the underspecified-vowel bundle are *not* deleted.

The reason is empirical. In this dataset, /j/ occurs in 30 inputs, but in 26 of these it is followed by fully specified vowels, which cannot supply the relevant contradictions

Φ	learned tier Θ	Matched σ
$\begin{bmatrix}\textbf{back}\\ \text{tap}\end{bmatrix}$	$\begin{bmatrix}+\\0\end{bmatrix}$	back V
	$\begin{bmatrix}-\\-\end{bmatrix}$	j
	$\begin{bmatrix}-\\0\end{bmatrix}$	front V
	$\begin{bmatrix}0\\0\end{bmatrix}$	underspecified V
$\begin{bmatrix}\textbf{front}\\ \text{tap}\end{bmatrix}$	$\begin{bmatrix}+\\-\end{bmatrix}$	j
	$\begin{bmatrix}+\\0\end{bmatrix}$	front V
	$\begin{bmatrix}-\\0\end{bmatrix}$	back V
	$\begin{bmatrix}0\\0\end{bmatrix}$	underspecified V
$\begin{bmatrix}\textbf{low}\\ \text{back}\\ \text{tap}\end{bmatrix}$	$\begin{bmatrix}+\\+\\0\end{bmatrix}$	ɑ
	$\begin{bmatrix}-\\+\\0\end{bmatrix}$	o, u, ɯ
	$\begin{bmatrix}-\\-\\-\end{bmatrix}$	j
	$\begin{bmatrix}-\\-\\0\end{bmatrix}$	i, y, e
	$\begin{bmatrix}0\\0\\0\end{bmatrix}$	E
$\begin{bmatrix}\textbf{round}\\ \text{back}\\ \text{tap}\\ \text{high}\end{bmatrix}$	$\begin{bmatrix}+\\+\\0\\+\end{bmatrix}$	u
	$\begin{bmatrix}+\\+\\0\\-\end{bmatrix}$	o
	$\begin{bmatrix}+\\-\\0\\+\end{bmatrix}$	y
	$\begin{bmatrix}-\\+\\0\\+\end{bmatrix}$	ɯ
	$\begin{bmatrix}-\\+\\0\\-\end{bmatrix}$	ɑ
	$\begin{bmatrix}-\\-\\-\\+\end{bmatrix}$	j
	$\begin{bmatrix}-\\-\\0\\+\end{bmatrix}$	i
	$\begin{bmatrix}-\\-\\0\\-\end{bmatrix}$	e
	$\begin{bmatrix}-\\0\\0\\-\end{bmatrix}$	E
	$\begin{bmatrix}0\\0\\0\\+\end{bmatrix}$	I

TABLE 7.3: Feature set Φ used to learn tiers for features ϕ (in bold) and tier content

needed for deletion. The remaining four items are the only cases that could, in principle, create informative contexts once consonants have been deleted from the tier and /j/ becomes adjacent (on tier) to an underspecified vowel. For example consider the following representative pairs:

/kojnI/ $\mapsto$ [kojnu] and /kojnE/ $\mapsto$ [kojnɑ].

Crucially, in these items the vowels preceding /j/ are all [+ back], and the same is true for the remaining two potentially informative examples. Consequently, the value for back assigned to the underspecified vowels will always be [+ back] as well. If, on the other hand, the dataset contained an otherwise identical pair in which the preceding vowel was front, that contrast would provide the appropriate double-witness evidence to delete /j/ from the tier. In the absence of such evidence, retaining /j/ is the expected outcome.

The same logic explains why the underspecified vowel bundle remains on the tier for [BACK] and [FRONT]. In this dataset, there is no independent opportunity for a tier element to be contradicted as it is never followed by another underspecified vowel. Once again, this should not be interpreted as an error. This behavior reflects a limitation of the observed sample, not of the learner. Importantly, the retention of underspecified-vowel bundles does not prevent correct learning of the harmony mapping for the given sample. It simply means that this is the safest tier hypothesis with the current evidence and it is consistent with the data.

Low For target [LOW], the successful feature set includes [BACK] and [TAP]. Here the induced partition is more fine-grained, and the underspecified non-high vowel (E) is appropriately distinguished as the only bundle that maps to a segment undergoing the relevant change. It is because [low] values never change for high vowels. The learner deletes bundles corresponding to consonants and I, while retaining vowel bundles and the approximant.

Round For target [ROUND], we use [BACK], [TAP], and [HIGH] in addition to [ROUND] itself. The resulting partition separates vowel classes quite finely, to the point where the learned tier may appear almost "segmental." Importantly, the featural representation still provides a computational advantage in tier inference because it compresses large consonant inventories (here 18 phonemes) into two distinct feature bundles. This allows the learner to eliminate irrelevant consonantal material more efficiently than it would under a purely segmental alphabet.

7.4.2 From tier content to a minimal deterministic transducer

To make the consequences of tier conservatism concrete, we now illustrate the canonical deterministic transducer construction for the [BACK] tier. Figure 7.3 shows the transducer built from the featural input alphabet under the learned tier for [BACK]. The sole off-tier element (here, the bundle $[\begin{smallmatrix}0\\-\end{smallmatrix}]$ corresponding to consonants) is implemented as self-loops at each state (excluding boundary states), reflecting that it does not update tier context. The remaining transitions correspond to the tier-sensitive contexts required by the canonical construction.

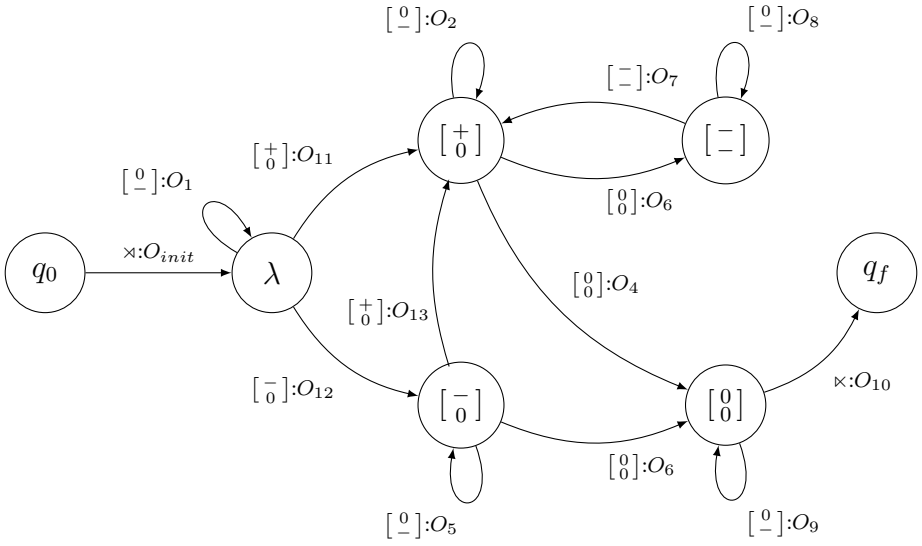


FIGURE 7.3: Deterministic transducer for [back] on tier.

At this point, the transducer *structure* is fixed by the tier and the window bound. The remaining question is how to assign concrete outputs to transitions. We do so using SEAL, exactly as in previous chapters. The representative encoded examples below illustrate the form of the resulting constraints and why the sample is sufficient for convergence in the present case.

$$O_1\, O_{11}\, O_3\, O_7\, O_4\, O_{10} \quad = \quad 0\,+\,-\,+\,+$$
$$O_1\, O_3\, O_9\, O_6\, O_5\, O_{14} \quad = \quad 0\,-\,+\,0\,+$$

In the L2R direction, the relevant environment for back harmony in Turkish is encountered before the target. Consequently, once a back or front vowel has been processed, the machine already determines the appropriate output value to produce when it later processes an underspecified vowel. The *push-right* variant SEAL, `PRD-SEAL`, can therefore resolve the resulting constraints using only the first stage of learning, i.e. the 1-to-1 alignment round. As a result, the non-identity transitions leading into the underspecified vowel are assigned the correct outputs: $O_4 = +$ and $O_6 = -$.

A notable byproduct of the learned tier for [BACK] is that /j/ remains on the tier. In the present dataset this does not compromise correctness, because there are no attested paths in which /j/ is followed by an underspecified vowel. Thus, retaining /j/ reflects the intended conservatism of `CanTIERr-2`. Potential difficulties may arise only if unseen paths are treated as identity transitions, as discussed in Chapter 6. We return to this issue in the Discussion section.

To conclude, this case study shows that the proposed input-oriented tier learner, `CanTIERr-2` can recover a correct tier for Turkish backness harmony from limited data. Given this tier (and the imposed memory bound $k = 2$), we then build the corresponding minimal subsequential transducer skeleton and apply SEAL to infer the transition outputs. In what follows is a discussion that discusses potential limitations of this approach and proposes future extensions.

To conclude, this case study shows that the proposed input-oriented tier learner, `CanTIERr-2`, can recover an appropriate tier for Turkish backness and roundness

harmonies from a limited dataset. Given this tier (and the resulting memory bound $k = 2$), we showed how to construct the corresponding minimal canonical subsequential transducer skeleton for each tier and how to apply SEAL to infer the transition outputs. The next section discusses limitations of this approach and outlines directions for future extensions.

7.5 Discussion and Conclusion

This chapter introduced a relaxed tier learner, `CanTIERr-2`, designed to behave conservatively in small samples while preserving the core learnability logic of Burness and McMullin (2019) for inferring tiers for 2-OTSL functions. The core modification is the *double-witness* requirement: a symbol θ is removed from the tier only when an inconsistency is supported by two distinct contexts. This delays irreversible eliminations until the learner has more evidence, and as a result returns a conservative superset tier rather than an overconfident (and potentially incorrect) minimal tier.

A second modification addresses a distinct small-sample issues that arises from `lcp`-based prefix estimation: when a prefix p has only one observed continuation, $\texttt{lcp}(p)$ can "run ahead" and absorb output material that should be licensed only *after* reading the next symbol. We mitigated this by capping the estimated prefix output to the length of the prefix, enforcing $|\texttt{lcp}(p)| = |p|$, and by restricting comparisons in `CanTIERr-2` to single-symbol contributions (via `first`). Together, these changes prevent undesired empty-string contributions and the resulting artificial inconsistencies that can lead to deleting incorrect elements from the tier.

The Turkish case study illustrates both the strengths and the intended conservative bias of the relaxed learner. The learned tiers are not always *minimal* in the sense of containing only the restricted ideal vowel classes. Nevertheless, the inferred tiers are linguistically salient given the adopted feature system and the distribution of contexts in the dataset, and are consistent with the sample.

Nevertheless, the current approach has some limitations. Processes that involve insertion or deletion pose particular challenges, in part because the relaxed learner intentionally prevents `lcp` from "looking ahead" beyond the current prefix (by enforcing $|\texttt{lcp}(p)| = |p|$). This restriction is well-motivated for purely assimilatory tier processes, but it interacts poorly with epenthesis, which is a real phenomenon in Turkish. Inserting a segment, as discussed in previous chapters, affects *every* feature in the system, including those projected to the tier. In particular, if a segment is inserted between a root-final vowel and an underspecified vowel of the following suffix, then the output increment attributed to that suffixal vowel is no longer length 1: it includes both the epenthetic segment and the vowel's own predicted output. This creates a mismatch with stems that end in a consonant, where no epenthesis occurs and the same underspecified vowel contributes only its own feature value. It is because consonants are invisible on the tier and there is no way to differentiate those two contexts. As a result, the learner observes incompatible "contributions" for the same underlying configuration, which can spuriously trigger inconsistency-based deletions under the current tier-learning approach. This is exactly the reason why we excluded from the sample the input-output pairs that exhibited this vowel hiatus resolution cases by inserting the approximant [j].

More generally, based on the studies in this dissertation it seems that insertion/deletion phenomena typically require either (i) richer representations that encode the relevant conditioning environments, or (ii) additional machinery that separates such processes from those purely assimilatory ones. This motivation is consistent with the broader architecture developed in earlier chapters: `LCD-SEAL` was introduced precisely as an alternative to `RPD-SEAL` for settings in which epenthesis or deletion appears. For the problem at hand, a natural first extension to address this limitation would be to enrich the input representation with morpheme-boundary information, so that epenthesis can be conditioned without forcing consonants to remain visible on the harmony tier.

A second potential issue is that some languages require *different* visibility conditions for different processes, and may therefore demand multiple tiers that are tracked in parallel. For example, Kikongo exhibits a case in which two long-distance dependencies apply simultaneously: a vowel-height alternation affecting the suffixal vowel (e.g. /i/ surfacing as [e]) and a nasal harmony affecting the suffixal consonant (e.g. /d/ surfacing as [n]) in forms such as /meng-idi/ $\rightarrow$ [meng-ene]. This pattern can be analyzed as a strongly target-specified 2-$OMTSL$ function, defined in Burness and McMullin (2020), in which different segments "track" different tiers. Particularly, the suffixal vowel /i/ consults a vowel tier T_v (containing all and only vowels) to determine its harmonic output, while the suffixal consonant /d/ consults a nasal tier T_n (containing nasal consonants) to determine its output. In this way, the two processes are computed in parallel, each with its own visibility condition, motivating multi-tier representation and inference (Burness and McMullin, 2021).

While patterns of this type can be challenging under a purely segmental representation, particularly because capturing multiple simultaneous long-distance dependencies may require moving to a more complex function class, they are much less problematic under a featural representation. In fact, this is precisely the kind of situation that exemplifies the effectiveness of FxF. Different features can be learned with different tier projections, so one projection can support the feature changes responsible for the $\mathrm{i} \rightarrow \mathrm{e}$ alternation (height harmony), while a separate projection can support the features responsible for the $\mathrm{d} \rightarrow \mathrm{n}$ alternation (nasal harmony). In this sense, the featural decomposition allows the two processes to be learned and computed in parallel without positing a dedicated multi-tier class of functions.

Next, it is worth emphasizing, as shown explicitly in Chapter 6, that completing a learned transducer by filling in sample-missing paths with identity transitions is not always the ultimate solution. In the Turkish example, retaining /j/ on tier is harmless precisely because the sample contains no attested tier-path in which /j/ is followed

by an underspecified vowel. However, if the implementation assumes totality and assigns identity outputs to unattested transitions, then unseen combinations can introduce spurious behavior, echoing the concerns raised in Chapter 6. For this reason, it is preferable to treat the *minimal* learned transducer as the most conservative hypothesis at the current stage of learning, rather than forcing a total function prematurely.

Importantly, a learned function being *partial* does not necessarily mean that it is uninformative. Even when restricted to paths present in the sample, the FxF framework can still generalize well because learning takes place over features: the transducer encodes the fact that *any* back (front) vowel provides the relevant context for assigning harmonic values to a following underspecified vowel, and that *any* consonant is off the tier. What is ruled out are only those transitions for which the learner has not yet encountered evidence (e.g. an approximant followed by an underspecified vowel on tier, or certain word-final configurations). Rather than interpreting these absences as an error, it might be beneficial view them simply as a stage of hypothesis formation. This *stage* perspective is compatible with the idea that children revise their generalizations as additional evidence accumulates, including cases where an initially productive generalization must be weakened or abandoned once it becomes inconsistent with the input distribution (Yang, 2016). In the present setting, keeping the transducer partial serves the same conservative function, which is explicit avoidance of committing to behavior on unattested paths until the data provide support for doing so.

Throughout this dissertation, featural representations have proven useful for reducing the input alphabet size, shrinking the required state space, and consequently yielding smaller transducers. In addition, decomposing representations into features can factor phonological processes into (multiple) feature-specific learning problems. This modularity is promising, however it requires further investigation. At the same time, these results also suggest that an adequate phonological representation may need to include additional kinds of information beyond segmental features, such as morpheme-boundary cues, and potentially higher-level prosodic structure (e.g. stress,

distinctions between phonological and intonational domains, or tone). Recent work on the later demonstrates that appropriate representational choices are beneficial for successful learning of tonal phonotactics (Li, 2025). Ultimately, a key goal for future work is to extend the present learning architecture so that it can accommodate such enriched representations, including the computational and learnability benefits of the featural approach.

Chapter 8

Conclusion

This dissertation builds a bridge between the theoretical program of subregular phonology and empirical modeling by integrating phonological representations, features and tiers, into grammatical-inference algorithms for subsequential transducers, and by developing new learners as well as adapting existing ones for sparse-data regimes. The resulting framework preserves the central appeal of the subregular approach: precise, formally restricted hypothesis spaces with clear computational commitments, while also enabling direct experimental evaluation on case studies drawn from natural languages. In this way, the dissertation gives equal weight to formal results about learner behavior and to the representational commitments of phonological theory, and shows that subregular restrictions function not only as analytical constraints but also as practical inductive biases that support sample-efficient learning in small-data settings.

Across Chapters 3–7, we developed a modular learning architecture grounded in deterministic subsequential transducers and subregular locality (in particular, k-ISL and its tier-based extensions). This architecture enriches classical grammatical inference with phonologically motivated representations such as features and tiers, without abandoning interpretability or computational restriction. The result is a family of learners whose hypotheses are explicit transducers, enabling both quantitative evaluation (e.g., sample efficiency) and qualitative diagnosis (e.g., where and why generalization fails).

Chapter 3 introduced the *Factor by Feature* (FxF) framework, which decomposes a

segmental mapping into a collection of feature-wise mappings, with one learner per feature dimension. By replacing segment symbols with feature-value symbols, FxF shrinks the effective input alphabet for each subproblem. Since the size of the canonical k-ISL transducer skeleton depends directly on alphabet size, this representational change reduces the induced state space and, in turn, the amount of data required for reliable inference. Empirically, FxF yields substantial gains in characteristic sample size relative to segment-based learning, especially for locally conditioned alternations that affect only a limited subset of features.

Chapter 4 showed that featural factorization naturally opens the door to additional parameters that can be adjusted *per feature*. We investigated two such parameters: the locality bound k and processing direction (left-to-right vs. right-to-left). Two lessons emerged. First, many features are faithful, and for them the best hypothesis is often the smallest possible machine, essentially an identity mapping, which can be learned with minimal evidence. Second, directionality interacts with locality because the conditioning information for an alternation may appear before or after the target of the change. When the learner processes strings in a direction that encounters the conditioning context before the target, it can often avoid output withholding and safely default to identity on unattested paths. In this way, direction choice functions as a principled inductive bias toward efficient inference rather than merely an implementational detail.

Chapter 5 introduced a novel learner, SEAL (String Equation Adaptive Learner). Given a parameter choice (including k, feature projections, and direction), SEAL fixes the canonical k-ISL transducer skeleton and treats transition outputs as unknown variables. Training examples are then compiled into a system of string equations: each input induces a path whose unknown transition outputs jointly define a candidate output string, which is constrained to match the observed output. Learning thus reduces to solving this system by assigning concrete substrings to transition variables.

Crucially, SEAL is guided by an *identity-first* bias: it begins with a conservative alignment that treats identity as the default and redistributes output material only when global inconsistency forces revision. This bias is conceptually related to *invariant transparency*, the view that learners do not posit abstract underlying representations unless there is an alternation that forces to do so (Ringe and Eska, 2013; Richter, 2018, 2021). In this sense, SEAL uses the same conservative bias when learning transducers: identity mappings are preserved unless the observed data force a non-identity analysis. This global, consistency-driven approach yields substantial gains in sample efficiency and robustness compared to the previously adapted learner, SOSFIA.

Chapter 6 moved beyond synthetic paradigms to sparse, more naturalistic datasets adapted from phonology textbooks and problem sets. These datasets are intentionally small, often involve larger inventories than in previous experiments, and frequently contain interacting processes. We show that SEAL reliably converges to transducers consistent with the observed pairs and, in many cases, exhibits the expected generalization behavior beyond the training sample. Moreover, the transparency of the learned transducers enables fine-grained diagnosis, reframing evaluation as a tool for identifying which representational and inferential methods require further improvements.

Some phonological dependencies cannot be captured by any bounded window on the input side, even with enriched feature sets, because the relevant generalization requires large portions of the string to be ignored. Chapter 7 therefore extended the dissertation's core strategy to long-distance processes via tier-based representations. Building on prior work on tier-based strictly local languages and functions, we developed an input-oriented tier-learning procedure (CANTIER-2) together with conservative relaxed variants designed specifically for small-data settings. We then showed that once tier content is inferred, the remaining task again is reduced to output assignment over a fixed transducer structure, for which SEAL can be reused. On an example from Turkish we show that the tier learner identifies phonologically well-formed and

empirically adequate tier content for backness and rounding dependencies, thereby bringing a long-distance pattern often analyzed in output-oriented terms back into an input-local format compatible with the overall framework.

In sum, the dissertation's results support three broader conclusions. First, feature-based representations are not only central to phonological analysis but also have direct consequences for learning: by shrinking input alphabets and factorizing hypothesis spaces, they can improve learnability and sample efficiency. Second, inductive biases, particularly the default-to-identity, informed choices of processing direction, and the locality restrictions embodied in k-ISL functions and their tier-based variants, are well matched to the structure of phonological alternations and can substantially reduce the evidence required for correct inference. Third, interpretability is not a luxury but a core requirement: because hypotheses are explicit transducers, both successes and failures are diagnosable, enabling iterative refinement and tighter connections between computational learning behavior and phonological explanation.

At the same time, several directions remain open. FxF relies on identifying useful feature sets for predicting each output feature, and in the worst case searching this space can be expensive. Incorporating structure into the feature system—for example, through feature geometry—may allow a more constrained and realistic traversal of candidate feature sets. Relatedly, as Chapter 6 shows, behavior on unseen paths can lead to problematic inferences: defaulting to identity may be safe in some regions and misleading in others, suggesting the need for a better theory of conservative extrapolation beyond observed paths. Finally, as discussed in Chapter 7, it may be necessary to introduce additional representational structure—such as syllable- or morpheme-level information—to determine which paths are linguistically expected and therefore should be learned. Finally, systematic comparisons with rule-based learners, together with evaluations on larger corpora (including child-directed speech), would further clarify what is gained and what is lost by committing to transducer-based hypothesis representations.

Overall, this dissertation argued that progress on phonological learning requires combining (i) formally restricted hypothesis spaces with clear typological consequences, (ii) structured representations motivated by phonological theory, and (iii) conservative learning biases suited to sparse exposure. The results across local alternations, textbook-style datasets, and tier-based long-distance dependencies support the view that much of phonological and morphophonological computation can be learned by strictly local mechanisms *once the representational space is appropriately structured*. More broadly, the dissertation provides a blueprint for making grammatical inference methods linguistically faithful and empirically useful in small-data conditions without sacrificing interpretability.

Bibliography

Albright, A. (2009). Modeling analogy as probabilistic grammar. In J. P. Blevins and J. Blevins (Eds.), *Analogy in Grammar*, pp. 185–213. Oxford University Press.

Albright, A. and B. Hayes (2002). Modeling English past tense intuitions with minimal generalization. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning (SIGPHON)*, pp. 58–69. Association for Computational Linguistics.

Albright, A. and B. Hayes (2003). Rules vs. analogy in English past tenses: A computational/experimental study. *Cognition 90*(2), 119–161.

Alderete, J. (2003). Dissimilation. In W. Frawley (Ed.), *International Encyclopedia of Linguistics* (2 ed.). Oxford and New York: Oxford University Press.

Archangeli, D. (1991). Syllabification and prosodic templates in Yawelmani. *Natural Language & Linguistic Theory 9*(2), 231–283.

Atkinson, J., F. W. Campbell, and M. R. Francis (1976). The magic number 4 +/- 0: A new look at visual numerosity judgements. *Perception 5*(3), 327–334.

Avery, P. and W. Idsardi (2001). Laryngeal dimensions, completion and enhancement. In T. A. Hall and U. Kleinhanz (Eds.), *Studies in Distinctive Feature Theory*, pp. 41–70. Berlin: Mouton de Gruyter.

Baković, E. (2007). A revised typology of opaque generalisations. *Phonology 24*, 217–259.

Baković, E. and L. Blumenfeld (2024, Apr.). A formal typology of process interactions. *Phonological Data and Analysis 6*(3), 1–43.

Bale, A. and C. Reiss (2018). *Phonology: A Formal Introduction*. The MIT Press.

Beesley, K. R. and L. Karttunen (2003). *Finite State Morphology*. CSLI Publications.

Beguš, G. (2020, January). Modeing unsupervised phonetic and phonological learning in generative adversarial phonology. In A. Ettinger, G. Jarosz, and J. Pater (Eds.), *Proceedings of the Society for Computation in Linguistics 2020*, New York, New York, pp. 38–48. Association for Computational Linguistics.

Belth, C. (2023). Identity and locality: Computational parsimony in phonological learning. *Phonology*. In press.

Belth, C. (2024, 08). A learning-based account of phonological tiers. *Linguistic Inquiry*, 1–37.

Berko, J. (1958). The child's learning of English morphology. *Word 14*, 150–177.

Bethin, C. (1978). *Phonological Rules in the Nominative Singular and the Genitive Plural of the Slavic Substantive Declension*. Ph.d. dissertation, University of Illinois at Urbana-Champaign.

Bhaskar, S., J. Chandlee, A. Jardine, and C. Oakden (2020). Boolean monadic recursive schemes as a logical characterization of the subsequential functions. In A. Leporati, C. Martín-Vide, D. Shapira, and C. Zandron (Eds.), *Language and Automata Theory and Applications - LATA 2020*, Lecture Notes in Computer Science, pp. 157–169. Springer.

Blevins, J. (2004). A reconsideration of Yokuts vowels. *International Journal of American Linguistics 70*(1), 33–51.

Blevins, J. P., P. Milin, and M. Ramscar (2017). The Zipfian paradigm cell filling problem. In F. Kiefer, J. P. Blevins, and H. Bartos (Eds.), *Perspectives on Morphological Structure: Data and Analyses*, pp. 139–158. Leiden: Brill.

Boersma, P. and B. Hayes (2001). Empirical tests of the gradual learning algorithm. *Linguistic Inquiry 32*(1), 45–86.

Brown, G. (1972). *Phonological Rules and Dialect Variation: A Study of the Phonology of Lumasaaba*. Ph. D. thesis, University of Edinburgh.

Browne, W. (1993). Serbo-Croat. In B. Comrie and G. G. Corbett (Eds.), *The Slavonic Languages*, pp. 306–387. London and New York: Routledge.

Buckley, E. (2001). Polish o-raising and phonological explanation. Manuscript, University of Pennsylvania.

Burness, P. and K. McMullin (2019, July). Efficient learning of output tier-based strictly 2-local functions. In P. de Groote, F. Drewes, and G. Penn (Eds.), *Proceedings of the 16th Meeting on the Mathematics of Language*, Toronto, Canada, pp. 78–90. Association for Computational Linguistics.

Burness, P. and K. McMullin (2020, July). Multi-tiered strictly local functions. In G. Nicolai, K. Gorman, and R. Cotterell (Eds.), *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Online, pp. 245–255. Association for Computational Linguistics.

Burness, P. and K. McMullin (2021, 23–27 Aug). Learning multiple independent tier-based processes. In J. Chandlee, R. Eyraud, J. Heinz, A. Jardine, and M. van Zaanen (Eds.), *Proceedings of the Fifteenth International Conference on Grammatical Inference*, Volume 153 of *Proceedings of Machine Learning Research*, pp. 66–80. PMLR.

Burness, P. A., K. J. McMullin, and J. Chandlee (2021). Long-distance phonological processes as tier-based strictly local functions. *Glossa: a journal of general linguistics 6*(1), 1–37.

Chan, E. (2008). *Structures and Distributions in Morphology Learning*. Ph. D. thesis, University of Pennsylvania.

Chandlee, J. (2014). *Strictly Local Phonological Processes*. Ph. D. thesis, University of Delaware.

Chandlee, J. (2017). Computational locality in morphological maps. *Morphology 27*(4), 599–641.

Chandlee, J., R. Eyraud, and J. Heinz (2014). Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics 2*, 491–504.

Chandlee, J., R. Eyraud, and J. Heinz (2015). Output strictly local functions. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, Chicago, USA, pp. 112–125. Association for Computational Linguistics.

Chandlee, J., R. Eyraud, J. Heinz, A. Jardine, and J. Rawski (2019, July). Learning with partially ordered representations. In P. de Groote, F. Drewes, and G. Penn (Eds.), *Proceedings of the 16th Meeting on the Mathematics of Language*, Toronto, Canada, pp. 91–101. Association for Computational Linguistics.

Chandlee, J. and J. Heinz (2018, January). Strict locality and phonological maps. *Linguistic Inquiry 49*(1), 23–60.

Chandlee, J., J. Heinz, and A. Jardine (2018). Input strictly local opaque maps. *Phonology 35*(2), 171–205.

Chandlee, J. and A. Jardine (2021). Computational universals in linguistic theory: Using recursive programs for phonological analysis. *Language 93*, 485–519.

Chen, M. Y. (1997). Acoustic correlates of English and French nasalized vowels. *Journal of the Acoustical Society of America 102*(4), 2360–2370.

Choffrut, C. (1977). Une caractérisation des fonctions séquentielles et des fonctions sous-séquentielles en tant que relations rationnelles. *Theoretical Computer Science 5*(3), 325–337.

Choffrut, C. (2003). Minimizing subsequential transducers: a survey. *Theoretical Computer Science 292*(1), 131 – 143.

Chomsky, N. (1986). *Knowledge of Language: Its Nature, Origin, and Use*. New York: Praeger.

Chomsky, N. and M. Halle (1965). Some controversial questions in phonological theory. *Journal of Linguistics 1*, 97–138.

Chomsky, N. and M. Halle (1986). *The Sound Pattern of English*. New York: Harper & Row.

Clements, G. N. (2009). The role of features in phonological inventories. In E. Raimy and C. E. Cairns (Eds.), *Contemporary Views on Architecture and Representations in Phonology*, pp. 19–68. Cambridge, MA: MIT Press.

Clements, G. N. and E. V. Hume (1995). The internal organization of speech sounds. In J. Goldsmith (Ed.), *The Handbook of Phonological Theory*, pp. 245–306. Cambridge, Mass.: Blackwell.

Clements, G. N. and E. Sezer (1982). Vowel and consonant disharmony in turkish. In H. van der Hulst and N. Smith (Eds.), *The Structure of Phonological Representations*, Volume Part II, pp. 213–256. Dordrecht: Foris Publications.

Cohn, A. C. (1993). Nasalisation in English: Phonology or phonetics. *Phonology 10*, 43–81.

Cohn, A. C. (2011). Features, segments, and the sources of phonological primitives. In R. Ridouane and G. N. Clements (Eds.), *Where do Phonological Contrasts Come From?*, Number 6 in Language Faculty and Beyond, pp. 15–41. Amsterdam: John Benjamins.

Colcombet, T. (2011). Green's relations and their use in automata theory. In *Language and Automata Theory and Applications*, Volume 6638 of *Lecture Notes in Computer Science*, Heidelberg, pp. 1–21. Springer-Verlag.

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences 24*(1), 87–114.

de la Higuera, C. (2010). *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press.

Dresher, B. E. (2009). *The Contrastive Hierarchy in Phonology*. Number 121 in Cambridge Studies in Linguistics. Cambridge: Cambridge University Press.

Du, N. and K. Durvasula (2025). *Psycholinguistics and Phonology: The Forgotten Foundations of Generative Phonology*. Cambridge University Press.

Duanmu, S. (2016). *A Theory of Phonological Features*. Oxford University Press.

Dunn, M. (1999). *A grammar of Chukchi*. Ph. D. thesis, Australian National University.

Ellis, K., A. Albright, A. Solar-Lezama, et al. (2022). Synthesizing theories of human language with bayesian program induction. *Nature Communications 13*, 5024.

Fodor, J. A. (1980). Fixation of belief and concept acquisition. In M. Piattelli-Palmarini (Ed.), *Language and Learning: The Debate Between Jean Piaget and Noam Chomsky*, pp. 142–162. Cambridge, MA: Harvard University Press.

Fowler, C. A. and J. M. Brown (2000). Perceptual parsing of acoustic consequences of velum lowering from information for vowels. *Perception & Psychophysics 62*(1), 21–32.

Fromkin, V. A. (1971). The non-anomalous nature of anomalous utterances. *Language 47*(1), 27–52.

Gerken, L. (2006). Decisions, decisions: Infant language learning when multiple generalizations are possible. *Cognition 98*(3), B67–B74.

Gildea, D. and D. Jurafsky (1996). Learning bias and phonological-rule induction. *Computational Linguistics 22*(4), 497–530.

Gold, E. M. (1967). Language identification in the limit. *Information and Control 10*(5), 447–474.

Goldsmith, J. (1976). *Autosegmental Phonology*. Ph. D. thesis, Massachusetts Institute of Technology.

Goldwater, S. and M. Johnson (2003). Learning OT constraint rankings using a maximum entropy model. In *Proceedings of the Workshop on Variation within Optimality Theory*, Stockholm, Sweden, pp. 111–120. Stockholm University.

Green, J. A. (1951). On the structure of semigroups. *Annals of Mathematics 54*(1), 163–172.

Gussmann, E. (2007). *The Phonology of Polish*. Oxford: Oxford University Press.

Halle, M. and G. N. Clements (1983). *Problem Book in Phonology: A Workbook for Introductory Courses in Linguistics and in Modern Phonology*. MIT Press.

Hansson, G. (2001). *Theoretical and Typological Issues in Consonant Harmony*. Ph. D. thesis, University of California, Berkeley.

Hayes, B. (2009). *Introductory Phonology*. Wiley-Blackwell.

Hayes, B. and C. Wilson (2008). A maximum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry 39*, 379–440.

Heinz, J. (2010a). Learning long-distance phonotactics. *Linguistic Inquiry 41*(4), 623–661.

Heinz, J. (2010b, July). String extension learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, pp. 897–906. Association for Computational Linguistics.

Heinz, J. (2011a). Computational phonology part I: Foundations. *Language and Linguistics Compass 5*(4), 140–152.

Heinz, J. (2011b). Computational phonology part II: Grammars, learning, and the future. *Language and Linguistics Compass 5*(4), 153–168.

Heinz, J. (2018). The computational nature of phonological generalizations. In L. Hyman and F. Plank (Eds.), *Phonological Typology*, Phonetics and Phonology, Chapter 5, pp. 126–195. De Gruyter Mouton.

Heinz, J. (2025). The logical structure of phonological generalizations. *Phonological Studies 28*, 69–80.

Heinz, J., C. de la Higuera, and M. van Zaanen (2015). *Grammatical Inference for Computational Linguistics*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.

Heinz, J., A. Kasprzik, and T. Kötzing (2012, October). Learning with lattice-structured hypothesis spaces. *Theoretical Computer Science 457*, 111–127.

Heinz, J. and R. Lai (2013). Vowel harmony and subsequentiality. In A. Kornai and M. Kuhlmann (Eds.), *Proceedings of the 13th Meeting on Mathematics of Language*, Sofia, Bulgaria.

Heinz, J., C. Rawal, and H. G. Tanner (2011). Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 58–64. Association for Computational Linguistics.

Heinz, J. and J. Sempere (Eds.) (2016). *Topics in Grammatical Inference*. Berlin Heidelberg: Springer-Verlag.

Hockett, C. F. (1967). The Yawelmani basic verb. *Language 43*(1), 208–222.

Hyman, L. (1975). *Phonology: Theory and Analysis*. Holt, Rinehart and Winston.

Jakobson, R., G. Fant, and M. Halle (1951). *Preliminaries to Speech Analysis*. Cambridge, MA: MIT Press.

Jardine, A. (2016). Computationally, tone is different. *Phonology 33*(2), 247–283.

Jardine, A., J. Chandlee, R. Eyraud, and J. Heinz (2014, September). Very efficient learning of structured classes of subsequential functions from positive data. In A. Clark, M. Kanazawa, and R. Yoshinaka (Eds.), *Proceedings of the Twelfth International Conference on Grammatical Inference (ICGI 2014)*, Volume 34, pp. 94–108. JMLR: Workshop and Conference Proceedings.

Jardine, A. and J. Heinz (2016). Learning tier-based strictly 2-local languages. *Transactions of the Association for Computational Linguistics 4*, 87–98.

Jardine, A. and K. McMullin (2017). Efficient learning of tier-based strictly k-local languages. In F. Drewes, C. Martín-Vide, and B. Truthe (Eds.), *Language and Automata Theory and Applications, 11th International Conference*, Lecture Notes in Computer Science, pp. 64–76. Springer.

Jarosz, G. (2011). Restrictiveness and phonological grammar and lexicon learning. *Lingua 121*(13), 2207–2230.

Johnson, C. D. (1972). *Formal aspects of phonological description*. Mouton.

Johnson, M. (1984, July). A discovery procedure for certain phonological rules. In *10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics*, Stanford, California, USA, pp. 344–347. Association for Computational Linguistics.

Kager, R. (1999). *Optimality Theory*. Cambridge University Press.

Kaplan, R. M. and M. Kay (1994). Regular models of phonological rules systems. *Computational Linguistics 20*, 331–378.

Kaye, J. (1980). The mystery of the tenth vowel. *Journal of Linguistic research 1*, 1–14.

Kenstowicz, M. (1994). *Phonology in Generative Grammar*. Blackwell Publishing.

Kenstowicz, M. and C. Kisseberth (1979). *Generative Phonology: Description and Theory*. Academic Press.

Kimball, G. D. (1991). *Koasati Grammar*. Studies in the Anthropology of North American Indians. Lincoln: University of Nebraska Press.

Kiparsky, P. (1973). Abstractness, opacity and global rules. In O. Fujimura (Ed.), *Three Dimensions of Linguistic Theory*, pp. 57–86. Tokyo: TEC. Part 2 of "Phonological representations".

Kodner, J. (2022). Language acquisition guiding theory and diachrony: A case study from latin morphology. *Natural Language & Linguistic Theory 41*, 733–792.

Kodner, J., S. Payne, S. Khalifa, and Z. Liu (2025). Evaluating learning trajectories of neural morphology acquisition models. *Linguistics Vanguard*.

Krakow, R. A. (1993). Nonsegmental influences on velum movement patterns: Syllables, sentences, stress, and speaking rate. In M. K. Huffman and R. A. Krakow (Eds.), *Nasals, nasalization, and the velum*, pp. 87–116. Academic Press.

Krakow, R. A. and M. K. Huffman (1989). Temporal coordination in speech production: The nasalization of vowels in English. *Journal of Phonetics 17*, 277–284.

Krause, S. R. (1980). *Topics in Chukchee Phonology and Morphology*. Ph. D. thesis, University of Illinois at Urbana-Champaign.

Krämer, M. (2015). *The Phonology of Vowel Harmony*. Cambridge University Press.

Kuo, J. (2024). Phonetic naturalness in the reanalysis of Samoan thematic consonant alternations. *Journal of Phonetics 107*, 101355. Special Issue of HISPhonCog.

Lambert, D. (2022). *Unifying Classification Schemes for Languages and Processes With Attention to Locality and Relativizations Thereof*. Ph. D. thesis, Stony Brook University.

Lambert, D. (2023). Relativized adjacency. *Journal of Logic, Language and Information 32*(4), 707–731.

Lambert, D. (2026). Multitier phonotactics with logic and algebra. *Phonology 43*, e9–31.

Lambert, D. and J. Heinz (2023). An algebraic characterization of total input strictly local functions. In *Proceedings of the Society for Computation in Linguistics*, Volume 6, pp. 25–34.

Lambert, D. and J. Heinz (2024). Algebraic reanalysis of phonological processes described as output-oriented. In *Proceedings of the Society for Computation in Linguistics*, Volume 7, pp. 129–138.

Lambert, D., J. Rawski, and J. Heinz (2021). Typology emerges from simplicity in representations and learning. *Journal of Language Modelling 9*(1), 151–194.

Lambert, D. and J. Rogers (2020). Tier-based strictly local stringsets: Perspectives from model and automata theory. In *Proceedings of the Society for Computation in Linguistics*, Volume 3, New Orleans, Louisiana, pp. 330–337.

Lamont, A. (2021). Optimizing over subsequences generates context-sensitive languages. *Transactions of the Association for Computational Linguistics 9*, 528–537.

Lamont, A. (2022). Optimality theory implements complex functions with simple constraints. *Phonology 38*(4), 729–740.

Li, H. (2025). Learning tonotactic patterns over autosegmental representations. *Proceedings of the Annual Meetings on Phonology 1*(1).

Lignos, C. and C. Yang (2016). *Morphology and Language Acquisition*, pp. 765–791. Cambridge Handbooks in Language and Linguistics. Cambridge University Press.

Luck, S. J. and E. K. Vogel (1997). The capacity of visual working memory for features and conjunctions. *Nature 390*(6657), 279–281.

MacWhinney, B. (2000). *The CHILDES Project: Tools for Analyzing Talk* (3 ed.). Mahwah, NJ: Lawrence Erlbaum Associates. CHILDES is supported by NICHD grant HD082736.

Marcus, G. F., S. Vijayan, S. Rao, and P. M. Vishton (1999). Rule learning by seven-month-old infants. *Science 283*(5398), 77–80.

Markowska, M., J. Heinz, and O. Rambow (2021, August). Finite-state model of shupamem reduplication. In G. Nicolai, K. Gorman, and R. Cotterell (Eds.), *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Online, pp. 212–221. Association for Computational Linguistics.

Mayer, C. (2020). An algorithm for learning phonological classes from distributional similarity. *Phonology 37*, 91–131.

McCollum, A. G., E. Baković, A. Mai, and E. Meinhardt (2020). Unbounded circumambient patterns in segmental phonology. *Phonology 37*(2), 215–255.

McMullin, K. (2016). *Tier-Based Locality in Long-Distance Phonotactics: Learnability and Typology*. Ph. D. thesis, University of British Columbia.

Mielke, J. (2004–2018). P-base: A database of phonological patterns. http://pbase.phon.chass.ncsu.edu/. Online database.

Mielke, J. (2008). *The Emergence of Distinctive Features*. Oxford University Press UK.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review 63*(2), 81–97.

Mohri, M. (1997). Finite-state transducers in language and speech processing. *Computational Linguistics 23*(2), 269–311.

Mosel, U. and E. Hovdhaugen (1992). *Samoan Reference Grammar*. Oslo: Scandinavian University Press.

Nag, A., S. Chakrabarti, A. Mukherjee, and N. Ganguly (2024). Efficient continual pre-training of LLMs for low-resource languages. *arXiv preprint arXiv:2412.10244*.

National Park Service (2025). History and the islands of samoa. Accessed 6 June 2026.

Nurse, D. and G. Philippson (Eds.) (2003). *The Bantu Languages*. London: Routledge.

Odden, D. (2005). *Introducing Phonology* (1 ed.). Cambridge Introductions to Language and Linguistics. Cambridge University Press.

Odden, D. (2014). *Introducing Phonology* (2nd ed.). Cambridge University Press.

Oncina, J. and P. Garcia (1991). Inductive learning of subsequential functions. Technical Report DSIC II-34, University Politécnia de Valencia.

Oncina, J., P. García, and E. Vidal (1993, May). Learning subsequential transducers for pattern recognition tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence 15*, 448–458.

Pisoni, D. B. (1977). Identification and discrimination of the relative onset time of two component tones: Implications for voicing perception in stops. *The Journal of the Acoustical Society of America 61*(5), 1352–1361.

Prickett, B. (2021). *Learning Phonology with Sequence-to-Sequence Neural Networks*. Ph. D. thesis, Univeristy of Massachusetts.

Prince, A. and P. Smolensky (1993). Optimality theory: Constraint interaction in generative grammar. *Technical Report 2 Rutgers University Center for Cognitive Science*.

Prince, A. and B. Tesar (2004). Fixing priorities: constraints in phonological acquisition. In *Fixing Priorities: Constraints in Phonological Acquisition*. Cambridge, MA: Cambride University Press.

Proctor, M., L. Goldstein, A. Lammert, D. Byrd, A. Toutios, and S. Narayanan (2013). Velic coordination in French nasals: A real-time magnetic resonance imaging study. In *Proceedings of Interspeech*, pp. 577–581.

Rasin, E., I. Berger, N. Lan, I. Shefi, and R. Katzir (2021). Approaching explanatory adequacy in phonology using minimum description length. *Journal of Language Modelling 9*(1), 17–66.

Rawski, J. (2021). *Structure and Learning in Natural Language*. Ph. D. thesis, Stony Brook University.

Reiss, C. (2012). Towards a bottom-up approach to phonological typology. In A. di Sciullo (Ed.), *Towards a Biolinguistic Understanding of Grammar*, pp. 169–191. John Benjamins, Philadelphia.

Reiss, C. and V. Volenec (2022). Conquer primal fear: Phonological features are innate and substance free. *Canadian Journal of Linguistics 67*(4), 581–610.

Richter, C. (2018). Learning allophones: What input is necessary? In A. B. Bertolini and M. J. Kaplan (Eds.), *Proceedings of the 42nd Annual Boston University Conference on Language Development*, Somerville, MA, pp. 659–672. Cascadilla Press.

Richter, C. L. (2021). *Alternation-Sensitive Phoneme Learning: Implications for Children's Development and Language Change*. Ph. D. thesis, University of Pennsylvania.

Ringe, D. and J. F. Eska (2013). *Historical Linguistics: Toward a Twenty-First Century Reintegration*. Cambridge: Cambridge University Press.

Roca, I. and W. Johnson (1991). *A Workbook in Phonology*. Blackwell.

Rogers, J., J. Heinz, M. Fero, J. Hurst, D. Lambert, and S. Wibel (2013). Cognitive and sub-regular complexity. In *Formal Grammar*, Volume 8036 of *Lecture Notes in Computer Science*, pp. 90–108. Springer.

Rogers, J. and G. K. Pullum (2011). Aural pattern recognition experiments and the subregular hierarchy. In C. Retoré (Ed.), *Logical Aspects of Computational Linguistics*, Volume 6735 of *Lecture Notes in Computer Science*, pp. 188–206. Springer.

Rose, S. and R. Walker (2004). A typology of consonant agreement as correspondence. *Language 80*(3), 475–531.

Rubach, J. (2019). Surface velar palatalization in Polish. *Natural Language & Linguistic Theory 37*, 1421–1462.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence 1*, 206–215.

Sakarovitch, J. (2009). *Elements of Automata Theory*. Cambridge University Press. Translated by Reuben Thomas from the 2003 edition published by Vuibert, Paris.

Selkirk, E. O. (1972). *The phrase phonology of English and French*. Ph. D. thesis, MIT.

Simonović, M. (2016). Nepostojano ili nepostojeće?: Generativni pristup nepostojanom a u srpskohrvatskom. In B. Arsenijević and S. Halupka-Rešetar (Eds.), *Srpski jezik u savremenoj lingvističkoj teoriji*, pp. 17–36. Niš: Filozofski fakultet, Univerzitet u Nišu.

Solé, M.-J. (1995). Spatio-temporal patterns of velo-pharyngeal action in phonetic and phonological nasalization. *Language and Speech 38*(1), 1–23.

Strother-Garcia, K., J. Heinz, and H. J. Hwangbo (2016, October). Using model theory for grammatical inference: a case study from phonology. In S. Verwer, M. van Zaanen, and R. Smetsers (Eds.), *Proceedings of The 13th International Conference on Grammatical Inference*, Volume 57 of *JMLR: Workshop and Conference Proceedings*, pp. 66–78.

Swanson, L., J. Heinz, and J. Rawski (2025). Phonotactic learning and constraint selection without statistics. *Linguistic Inquiry*. Accepted subject to minor revisions.

Tesar, B. and P. Smolensky (1998). Learnability in optimality theory. *Linguistic Inquiry* (29), 229–268.

Tesar, B. and P. Smolensky (2000). *Learnability in Optimality Theory*. Cambridge, MA: MIT Press.

Trubetzkoy, N. S. (1969). *Principles of Phonology*. Berkeley: University of California Press. Translation of *Grundzüge der Phonologie*.

Valiant, L. G. (2013). *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World*. New York: Basic Books.

Vihman, M. M. (1996). *Phonological Development: The Origins of Language in the Child*. Blackwell.

Vu, M. H., A. Zehfroosh, K. Strother-Garcia, M. Sebok, J. Heinz, and H. G. Tanner (2018, July). Statistical relational learning with unconventional string models. *Frontiers in Robotics and AI 5*(76), 1–26.

Wieczorek, W. (2017). *Grammatical Inference: Algorithms, Routines and Applications*. Springer.

Wiemerslage, A., A. Mueller, G. Nicolai, M. Silfverberg, and D. Yarowsky (2022). Morphological processing of low-resource languages: Where we are and what's next. *arXiv preprint arXiv:2203.08909*.

Yang, C. (2016). *The Price of Linguistic Productivity: How Children Learn to Break the Rules of Language*. Cambridge, MA: MIT Press.

Yang, C. D. (2006). *The Infinite Gift: How Children Learn and Unlearn the Languages of the World*. New York, NY, USA: Scribner.

Yang, C. M. and K. Ellis (2021). Phonological interactions, process types, and minimum description length principles. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Volume 43.

Yolyan, T. (2025a). A framework for learning phonological maps as logical transductions. In *Proceedings of the 18th Meeting of the Mathematics of Language*. Forthcoming.

Yolyan, T. (2025b). *Phonological Expressivity and Learning via Boolean Monadic Recursive Schemes*. Ph. D. thesis, Rutgers University.

Zsiga, E. (2013). *The Sounds of Language: An Introduction to Phonetics and Phonology*. Wiley-Blackwell.

Zuraw, K. (2000). *Patterned Exceptions in Phonology*. Ph. D. thesis, University of California, Los Angeles.

Zuraw, K., K. M. Yu, and R. Orfitelli (2014). The word-level prosody of samoan. *Phonology 31*(2), 271–327.

Appendix A

A.1 Machines and their sizes

Feature set Φ	$\lvert\mathbf{Q}\rvert$	$\lvert\mathbf{\Sigma_\Phi}\rvert$	$\lvert\delta\rvert$	$\neg$ **faith.** $\lvert\delta\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\neg$ **faith.** $\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$
[cons]	5	2	10	0	14	0
[son]	5	2	10	0	14	0
[cont]	5	2	10	0	14	0
[**nasal**, cons]	6	3	17	6	39	7
[approx]	6	3	17	0	39	0
[strid]	6	3	17	0	39	0
[lat]	6	3	17	0	39	0
[lab]	6	3	17	0	39	0
[labdent]	6	3	17	0	39	0
[cor]	6	3	17	0	39	0
[dor]	6	3	17	0	39	0
[ant]	6	3	17	0	39	0
[voice]	5	2	10	0	14	0
[round]	5	2	10	0	14	0
[high]	6	3	17	0	39	0
[low]	6	3	17	0	39	0
[tense]	6	3	17	0	39	0
[front]	6	3	17	0	39	0
[back]	6	3	17	0	39	0
[long]	5	2	10	0	14	0
[delrel]	6	3	17	0	39	0
[distr]	6	3	17	0	39	0
SUM:	126	60	332	6	701	7

TABLE A.1: English L2R transducer statistics for each feature or feature combination for $k = 2$.

Feature set Φ	$\lvert\mathbf{Q}\rvert$	$\lvert\mathbf{\Sigma_\Phi}\rvert$	$\lvert\delta\rvert$	$\neg$ **faith.** $\lvert\delta\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\neg$ **faith.** $\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$
[cons]	5	2	10	0	14	0
[son]	5	2	10	0	14	0
[cont]	5	2	10	0	14	0
[**nasal**, cons]	6	3	17	1	39	7
[approx]	6	3	17	0	39	0
[strid]	6	3	17	0	39	0
[lat]	6	3	17	0	39	0
[lab]	6	3	17	0	39	0
[labdent]	6	3	17	0	39	0
[cor]	6	3	17	0	39	0
[dor]	6	3	17	0	39	0
[ant]	6	3	17	0	39	0
[voice]	5	2	10	0	14	0
[round]	5	2	10	0	14	0
[high]	6	3	17	0	39	0
[low]	6	3	17	0	39	0
[tense]	6	3	17	0	39	0
[front]	6	3	17	0	39	0
[back]	6	3	17	0	39	0
[long]	5	2	10	0	14	0
[delrel]	6	3	17	0	39	0
[distr]	6	3	17	0	39	0
SUM:	126	60	332	1	701	7

TABLE A.2: English R2L transducer statistics for each feature or feature combination for $k = 2$.

Feature set Φ	$\lvert\mathbf{Q}\rvert$	$\lvert\mathbf{\Sigma_\Phi}\rvert$	$\lvert\delta\rvert$	$\neg$ **faith.** $\lvert\delta\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\neg$ **faith.** $\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	**k**
[cons]	3	2	4	0	2	0	1
[son]	3	2	4	0	2	0	1
[cont]	3	2	4	0	2	0	1
[**nasal**, cons]	6	3	17	6	39	7	2
[approx]	3	3	5	0	3	0	1
[strid]	3	3	5	0	3	0	1
[lat]	3	3	5	0	3	0	1
[lab]	3	3	5	0	3	0	1
[labdent]	3	3	5	0	3	0	1
[cor]	3	3	5	0	3	0	1
[dor]	3	3	5	0	3	0	1
[ant]	3	3	5	0	3	0	1
[voice]	3	2	4	0	2	0	1
[round]	3	2	4	0	2	0	1
[high]	3	3	5	0	3	0	1
[low]	3	3	5	0	3	0	1
[tense]	3	3	5	0	3	0	1
[front]	3	3	5	0	3	0	1
[back]	3	3	5	0	3	0	1
[long]	3	2	4	0	2	0	1
[delrel]	3	3	5	0	3	0	1
[distr]	3	3	5	0	3	0	1
SUM:	69	60	116	6	96	7	

TABLE A.3: English L2R transducer statistics for each feature or feature combination for learned k.

Feature set Φ	$\lvert\mathbf{Q}\rvert$	$\lvert\mathbf{\Sigma_\Phi}\rvert$	$\lvert\delta\rvert$	$\neg$ **faith.** $\lvert\delta\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\neg$ **faith.** $\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	**k**
[cons]	3	2	4	0	2	0	1
[son]	3	2	4	0	2	0	1
[cont]	3	2	4	0	2	0	1
[**nasal**, cons]	6	3	17	1	39	7	2
[approx]	3	3	5	0	3	0	1
[strid]	3	3	5	0	3	0	1
[lat]	3	3	5	0	3	0	1
[lab]	3	3	5	0	3	0	1
[labdent]	3	3	5	0	3	0	1
[cor]	3	3	5	0	3	0	1
[dor]	3	3	5	0	3	0	1
[ant]	3	3	5	0	3	0	1
[voice]	3	2	4	0	2	0	1
[round]	3	2	4	0	2	0	1
[high]	3	3	5	0	3	0	1
[low]	3	3	5	0	3	0	1
[tense]	3	3	5	0	3	0	1
[front]	3	3	5	0	3	0	1
[back]	3	3	5	0	3	0	1
[long]	3	2	4	0	2	0	1
[delrel]	3	3	5	0	3	0	1
[distr]	3	3	5	0	3	0	1
SUM:	69	60	116	1	96	7	

TABLE A.4: English R2L transducer statistics for each feature or feature combination for learned k.

Feature set Φ	$\lvert\mathbf{Q}\rvert$	$\lvert\mathbf{\Sigma_\Phi}\rvert$	$\lvert\delta\rvert$	$\neg$ **faith.** $\lvert\delta\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\neg$ **faith.** $\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$
[high]	15	3	53	0	363	0
[low]	9	2	22	0	62	0
[tense]	15	3	53	0	363	0
[front]	15	3	53	0	363	0
[back]	15	3	53	0	363	0
[round]	9	2	22	0	62	0
[long]	15	3	53	24	363	34
[cons]	9	2	22	0	62	0
[son]	9	2	22	0	62	0
[cont]	9	2	22	0	62	0
[delrel]	9	2	22	0	62	0
[approx]	9	2	22	0	62	0
[nasal]	9	2	22	0	54	0
[voice]	9	2	22	0	54	0
[lab]	9	2	22	0	62	0
[labdent]	9	2	22	0	62	0
[cor]	15	3	53	0	336	0
[ant]	15	3	53	0	309	0
[distr]	9	2	22	0	62	0
[strid]	9	2	22	0	62	0
[lat]	9	2	22	0	62	0
[dor]	15	3	53	0	336	0
[constrgl]	15	3	53	0	336	0
[spreadgl]	9	2	22	0	62	0
SUM:	270	57	807	24	4,028	34

TABLE A.5: L2R transducer statistics for each feature for the Yawelmani process for $k = 3$.

Feature set Φ	$\lvert\mathbf{Q}\rvert$	$\lvert\mathbf{\Sigma_\Phi}\rvert$	$\lvert\delta\rvert$	$\neg$ **faith.** $\lvert\delta\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\neg$ **faith.** $\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$
[high]	15	3	53	0	363	0
[low]	9	2	22	0	62	0
[tense]	15	3	53	0	363	0
[front]	15	3	53	0	363	0
[back]	15	3	53	0	363	0
[round]	9	2	22	0	62	0
[long]	15	3	53	1	363	34
[cons]	9	2	22	0	62	0
[son]	9	2	22	0	62	0
[cont]	9	2	22	0	62	0
[delrel]	9	2	22	0	62	0
[approx]	9	2	22	0	62	0
[nasal]	9	2	22	0	54	0
[voice]	9	2	22	0	54	0
[lab]	9	2	22	0	62	0
[labdent]	9	2	22	0	62	0
[cor]	15	3	53	0	336	0
[ant]	15	3	53	0	309	0
[distr]	9	2	22	0	62	0
[strid]	9	2	22	0	62	0
[lat]	9	2	22	0	62	0
[dor]	15	3	53	0	336	0
[constrgl]	15	3	53	0	336	0
[spreadgl]	9	2	22	0	62	0
SUM:	270	57	807	1	4,028	34

TABLE A.6: R2L transducer statistics for each feature for the Yawelmani process for $k = 3$.

Feature set Φ	$\lvert\mathbf{Q}\rvert$	$\lvert\mathbf{\Sigma_\Phi}\rvert$	$\lvert\delta\rvert$	$\neg$ **faith.** $\lvert\delta\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\neg$ **faith.** $\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	**k**
[high]	3	3	5	0	3	0	1
[low]	3	2	4	0	2	0	1
[tense]	3	3	5	0	3	0	1
[front]	3	3	5	0	3	0	1
[back]	3	3	5	0	3	0	1
[round]	3	2	4	0	2	0	1
[long]	15	3	53	24	363	34	3
[cons]	3	2	4	0	2	0	1
[son]	3	2	4	0	2	0	1
[cont]	3	2	4	0	2	0	1
[delrel]	3	2	4	0	2	0	1
[approx]	3	2	4	0	2	0	1
[nasal]	3	2	4	0	2	0	1
[voice]	3	2	4	0	2	0	1
[lab]	3	2	4	0	2	0	1
[labdent]	3	2	4	0	2	0	1
[cor]	3	3	5	0	3	0	1
[ant]	3	3	5	0	3	0	1
[distr]	3	2	4	0	2	0	1
[strid]	3	2	4	0	2	0	1
[lat]	3	2	4	0	2	0	1
[dor]	3	3	5	0	3	0	1
[constrgl]	3	3	5	0	3	0	1
[spreadgl]	3	2	4	0	2	0	1
SUM:	84	57	153	24	417	34	

TABLE A.7: L2R transducer statistics for each feature for the Yawelmani process for learned k.

Feature set Φ	$\lvert\mathbf{Q}\rvert$	$\lvert\mathbf{\Sigma_\Phi}\rvert$	$\lvert\delta\rvert$	$\neg$ **faith.** $\lvert\delta\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\neg$ **faith.** $\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	**k**
[high]	3	3	5	0	3	0	1
[low]	3	2	4	0	2	0	1
[tense]	3	3	5	0	3	0	1
[front]	3	3	5	0	3	0	1
[back]	3	3	5	0	3	0	1
[round]	3	2	4	0	2	0	1
[long]	15	3	53	1	363	34	3
[cons]	3	2	4	0	2	0	1
[son]	3	2	4	0	2	0	1
[cont]	3	2	4	0	2	0	1
[delrel]	3	2	4	0	2	0	1
[approx]	3	2	4	0	2	0	1
[nasal]	3	2	4	0	2	0	1
[voice]	3	2	4	0	2	0	1
[lab]	3	2	4	0	2	0	1
[labdent]	3	2	4	0	2	0	1
[cor]	3	3	5	0	3	0	1
[ant]	3	3	5	0	3	0	1
[distr]	3	2	4	0	2	0	1
[strid]	3	2	4	0	2	0	1
[lat]	3	2	4	0	2	0	1
[dor]	3	3	5	0	3	0	1
[constrgl]	3	3	5	0	3	0	1
[spreadgl]	3	2	4	0	2	0	1
SUM:	84	57	153	1	417	34	

TABLE A.8: R2L transducer statistics for each feature for the Yawelmani process for learned k.

Feature set Φ	$\lvert\mathbf{Q}\rvert$	$\lvert\mathbf{\Sigma_\Phi}\rvert$	$\lvert\delta\rvert$	$\neg$ **faith.** $\lvert\delta\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\neg$ **faith.** $\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$
[high]	15	3	53	6	309	40
[low]	9	2	22	2	62	15
[tense]	9	2	22	4	62	15
[front]	15	3	53	6	309	40
[back]	15	3	53	6	309	40
[**round**, high]	15	3	53	6	309	40
[long]	9	2	22	4	62	15
[cons]	9	2	22	4	62	15
[**son**, low]	15	3	53	20	336	142
[**cont**, tense]	15	3	53	20	363	160
[**delrel**, long]	23	4	106	54	1,236	669
[approx]	9	2	22	4	62	15
[**nasal**, son]	15	3	53	20	336	142
[**voice**, cons]	15	3	53	20	363	160
[lab]	15	3	53	20	363	160
[labdent]	9	2	22	4	62	15
[cor]	15	3	53	20	363	160
[**ant**, lat]	23	4	106	54	1,236	669
[**distr**, cor]	23	4	106	54	1,236	669
[**strid**, lat]	23	4	106	54	1,236	669
[lat]	15	3	53	20	336	142
[dor]	9	2	22	4	62	15
SUM:	320	63	1,161	755	9,074	4,007

TABLE A.9: L2R transducer statistics for each feature or feature combination for the Chukchi process for $k = 3$.

Feature set Φ	$\lvert\mathbf{Q}\rvert$	$\lvert\mathbf{\Sigma_\Phi}\rvert$	$\lvert\delta\rvert$	$\neg$ **faith.** $\lvert\delta\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\neg$ **faith.** $\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$
[high]	15	3	53	1	309	40
[low]	9	2	22	1	62	15
[tense]	9	2	22	1	62	15
[front]	15	3	53	1	309	40
[back]	15	3	53	1	309	40
[**round**, high]	15	3	53	1	309	40
[long]	9	2	22	1	62	15
[cons]	9	2	22	1	62	15
[**son**, low]	15	3	53	4	336	142
[**cont**, tense]	15	3	53	4	363	160
[**delrel**, long]	23	4	106	9	1,236	669
[approx]	9	2	22	1	62	15
[**nasal**, son]	15	3	53	4	336	142
[**voice**, cons]	15	3	53	4	363	160
[lab]	15	3	53	4	363	160
[labdent]	9	2	22	1	62	15
[cor]	15	3	53	4	363	160
[**ant**, lat]	23	4	106	9	1,236	669
[**distr**, cor]	23	4	106	9	1,236	669
[**strid**, lat]	23	4	106	9	1,236	669
[lat]	15	3	53	4	336	142
[dor]	9	2	22	1	62	15
SUM:	320	63	1,161	75	9,074	4,007

TABLE A.10: R2L transducer statistics for each feature or feature combination for the Chukchi process for $k = 3$.

Feature set Φ	$\lvert\mathbf{Q}\rvert$	$\lvert\mathbf{\Sigma_\Phi}\rvert$	$\lvert\delta\rvert$	$\neg$ **faith.** $\lvert\delta\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\neg$ **faith.** $\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$
[voice, son]	15	3	53	21	336	112
[son]	9	2	22	0	62	0
[cons]	9	2	22	0	62	0
[cont]	9	2	22	0	62	0
[cor]	9	2	22	0	62	0
[ant]	15	3	53	0	336	0
[dor]	9	2	22	0	62	0
[lab]	9	2	22	0	62	0
[distr]	9	2	22	0	62	0
[delrel]	15	3	53	0	336	0
[nasal]	9	2	22	0	54	0
[lat]	9	2	22	0	62	0
[high, voice, son, low]	45	6	302	143	8,250	1,375
[back]	9	2	22	0	62	0
[low]	15	3	53	0	336	0
[tense, voice, son, round]	45	6	302	143	8,250	1,375
[round]	15	3	53	0	336	0
SUM:	255	47	1,089	307	18,792	2,862

TABLE A.11: L2R transducer statistics for each feature or feature combination for the Polish process for $k = 3$.

Feature set Φ	$\lvert\mathbf{Q}\rvert$	$\lvert\mathbf{\Sigma_\Phi}\rvert$	$\lvert\delta\rvert$	$\neg$ **faith.** $\lvert\delta\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\neg$ **faith.** $\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$
[voice, son]	15	3	53	1	336	112
[son]	9	2	22	0	62	0
[cons]	9	2	22	0	62	0
[cont]	9	2	22	0	62	0
[cor]	9	2	22	0	62	0
[ant]	15	3	53	0	336	0
[dor]	9	2	22	0	62	0
[lab]	9	2	22	0	62	0
[distr]	9	2	22	0	62	0
[delrel]	15	3	53	0	336	0
[nasal]	9	2	22	0	54	0
[lat]	9	2	22	0	62	0
[high, voice, son, low]	45	6	302	2	8,250	1,375
[back]	9	2	22	0	62	0
[low]	15	3	53	0	336	0
[tense, voice, son, round]	45	6	302	2	8,250	1,375
[round]	15	3	53	0	336	0
SUM:	255	47	1,089	5	18,792	2,862

TABLE A.12: R2L transducer statistics for each feature or feature combination for the Polish process for $k = 3$.

Feature set Φ	$\lvert\mathbf{Q}\rvert$	$\lvert\mathbf{\Sigma_\Phi}\rvert$	$\lvert\delta\rvert$	$\neg$ **faith.** $\lvert\delta\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\neg$ **faith.** $\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	**k**
[**voice**, son]	6	3	17	2	39	13	2
[son]	3	2	4	0	2	0	1
[cons]	3	2	4	0	2	0	1
[cont]	3	2	4	0	2	0	1
[cor]	3	2	4	0	2	0	1
[ant]	3	3	5	0	3	0	1
[dor]	3	2	4	0	2	0	1
[lab]	3	2	4	0	2	0	1
[distr]	3	2	4	0	2	0	1
[delrel]	3	3	5	0	3	0	1
[nasal]	3	2	4	0	2	0	1
[lat]	3	2	4	0	2	0	1
[**high**, voice, son, low]	45	6	302	143	8,250	1,375	3
[back]	3	2	4	0	2	0	1
[low]	3	3	5	0	3	0	1
[**tense**, voice, son, round]	45	6	302	143	8,250	1,375	3
[round]	3	3	5	0	3	0	1
SUM:	138	47	681	292	16,571	2,763	

TABLE A.13: L2R transducer statistics for each feature or feature combination for the Polish process for learned k.

Feature set Φ	$\lvert\mathbf{Q}\rvert$	$\lvert\mathbf{\Sigma_\Phi}\rvert$	$\lvert\delta\rvert$	$\neg$ **faith.** $\lvert\delta\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\neg$ **faith.** $\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	**k**
[**voice**, son]	6	3	17	1	39	13	2
[son]	3	2	4	0	2	0	1
[cons]	3	2	4	0	2	0	1
[cont]	3	2	4	0	2	0	1
[cor]	3	2	4	0	2	0	1
[ant]	3	3	5	0	3	0	1
[dor]	3	2	4	0	2	0	1
[lab]	3	2	4	0	2	0	1
[distr]	3	2	4	0	2	0	1
[delrel]	3	3	5	0	3	0	1
[nasal]	3	2	4	0	2	0	1
[lat]	3	2	4	0	2	0	1
[**high**, voice, son, low]	45	6	302	2	8,250	1,375	3
[back]	3	2	4	0	2	0	1
[low]	3	3	5	0	3	0	1
[**tense**, voice, son, round]	45	6	302	2	8,250	1,375	3
[round]	3	3	5	0	3	0	1
SUM:	138	47	681	5	16,571	2,763	

TABLE A.14: R2L transducer statistics for each feature or feature combination for the Polish process for learned k.

Appendix B

B.1 English

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[$\substack{\textbf{nasal}\\ \text{cons}}$]	37	34	24	31	31	33	32	25	30	30
[**high**]	19	16	30	20	23	28	29	24	24	25
[**low**]	19	35	27	23	26	12	30	23	26	11
[**tense**]	32	29	27	18	33	33	3	21	32	23
[**front**]	21	32	6	23	8	26	23	19	27	19
[**back**]	29	30	17	26	22	22	17	20	34	24
[**round**]	11	8	8	12	8	7	6	5	8	11
[**long**]	5	7	7	9	10	7	7	9	11	9
[**cons**]	9	9	8	12	10	10	5	8	7	11
[**son**]	10	12	12	9	10	7	9	11	8	11
[**cont**]	11	8	5	11	12	10	13	3	6	9
[**delrel**]	25	20	17	23	28	33	28	28	21	30
[**approx**]	33	27	29	22	23	35	31	29	2	24
[**voice**]	7	10	11	9	6	13	6	10	5	11
[**lab**]	28	28	31	23	23	26	16	24	17	23
[**labdent**]	19	38	28	27	28	21	34	24	16	21
[**cor**]	14	25	22	18	21	25	22	32	23	25
[**ant**]	21	16	22	18	27	23	35	27	28	18
[**distr**]	29	26	31	19	27	26	21	26	31	24
[**strid**]	30	33	3	26	25	24	31	20	29	25
[**lat**]	23	26	23	26	32	23	33	27	32	31
[**dor**]	24	24	25	32	21	26	35	23	16	28
$\lvert\mathbf{M}_F\rvert$	189	206	177	179	196	198	190	174	184	186

TABLE B.1: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for the *insufficient* feature, across 10 runs, for VN in English with set $k = 2$, left-to-right.

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
$\left[\begin{smallmatrix}\textbf{nasal}\\ \text{cons}\end{smallmatrix}\right]$	37	34	33	37	35	12	33	31	31	9
[**high**]	29	22	12	10	24	24	25	10	27	4
[**low**]	20	27	4	4	21	4	30	4	4	4
[**tense**]	4	5	29	4	13	4	4	27	32	5
[**front**]	6	5	4	29	33	30	24	25	24	27
[**back**]	4	4	34	4	25	27	27	6	11	26
[**round**]	6	4	3	6	11	3	3	12	6	11
[**long**]	12	3	11	3	12	3	10	3	8	3
[**cons**]	3	6	6	3	7	8	4	8	3	3
[**son**]	10	13	9	13	7	13	4	3	3	10
[**cont**]	13	12	8	12	4	3	3	4	3	3
[**delrel**]	19	7	29	28	31	33	4	29	31	7
[**approx**]	30	31	25	4	4	24	4	4	4	34
[**voice**]	12	3	4	10	3	10	12	12	7	4
[**lab**]	5	28	31	8	23	4	15	13	28	4
[**labdent**]	4	4	4	25	25	18	4	7	30	25
[**cor**]	4	27	20	19	23	4	4	33	4	24
[**ant**]	4	31	24	4	29	9	34	4	5	17
[**distr**]	27	4	15	29	28	22	4	27	34	26
[**strid**]	4	17	19	28	28	32	23	4	13	4
[**lat**]	4	29	6	4	4	23	31	26	29	22
[**dor**]	28	4	26	6	13	33	4	27	4	23
$\lvert\mathbf{M}_F\rvert$	120	130	153	128	163	144	130	136	154	114

TABLE B.2: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for the *insufficient* feature, across 10 runs, for VN in English with set $k = 2$, right-to-left.

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
$[\,^{\textbf{nasal}}_{\textbf{cons}}\,]$	34	28	30	26	38	29	38	30	28	35
[**high**]	2	2	2	2	2	2	2	2	2	2
[**low**]	3	3	3	3	3	3	3	3	3	3
[**tense**]	3	3	3	3	3	3	3	3	3	3
[**front**]	3	3	3	3	3	3	3	3	3	3
[**back**]	3	3	3	3	3	3	3	3	3	3
[**round**]	2	2	2	2	2	2	2	2	2	2
[**long**]	2	2	2	2	2	2	2	2	2	2
[**cons**]	2	2	2	2	2	2	2	2	2	2
[**son**]	2	2	2	2	2	2	2	2	2	2
[**cont**]	2	2	2	2	2	2	2	2	2	2
[**delrel**]	3	3	3	3	3	3	3	3	3	3
[**approx**]	3	3	3	3	3	3	3	3	3	3
[**voice**]	2	2	2	2	2	2	2	2	2	2
[**lab**]	3	3	3	3	3	3	3	3	3	3
[**labdent**]	3	3	3	3	3	3	3	3	3	3
[**cor**]	3	3	3	3	3	3	3	3	3	3
[**ant**]	3	3	3	3	3	3	3	3	3	3
[**distr**]	3	3	3	3	3	3	3	3	3	3
[**strid**]	3	3	3	3	3	3	3	3	3	3
[**lat**]	3	3	3	3	3	3	3	3	3	3
[**dor**]	3	3	3	3	3	3	3	3	3	3
$\lvert\mathbf{M}_F\rvert$	51	41	46	41	49	45	51	48	43	45

TABLE B.3: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for the *insufficient* feature, across 10 runs, for VN in English with learned k, left-to-right.

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
$[\,^{\textbf{nasal}}_{\text{cons}}\,]$	34	22	32	36	32	32	31	35	29	32
[**high**]	2	2	2	2	2	2	2	2	2	2
[**low**]	3	3	3	3	3	3	3	3	3	3
[**tense**]	3	3	3	3	3	3	3	3	3	3
[**front**]	3	3	3	3	3	3	3	3	3	3
[**back**]	3	3	3	3	3	3	3	3	3	3
[**round**]	2	2	2	2	2	2	2	2	2	2
[**long**]	2	2	2	2	2	2	2	2	2	2
[**cons**]	2	2	2	2	2	2	2	2	2	2
[**son**]	2	2	2	2	2	2	2	2	2	2
[**cont**]	2	2	2	2	2	2	2	2	2	2
[**delrel**]	3	3	3	3	3	3	3	3	3	3
[**approx**]	3	3	3	3	3	3	3	3	3	3
[**voice**]	2	2	2	2	2	2	2	2	2	2
[**lab**]	3	3	3	3	3	3	3	3	3	3
[**labdent**]	3	3	3	3	3	3	3	3	3	3
[**cor**]	3	3	3	3	3	3	3	3	3	3
[**ant**]	3	3	3	3	3	3	3	3	3	3
[**distr**]	3	3	3	3	3	3	3	3	3	3
[**strid**]	3	3	3	3	3	3	3	3	3	3
[**lat**]	3	3	3	3	3	3	3	3	3	3
[**dor**]	3	3	3	3	3	3	3	3	3	3
$\lvert\mathbf{M}_F\rvert$	44	41	42	45	45	44	44	48	44	43

TABLE B.4: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for the *insufficient* feature, across 10 runs, for VN in English with learned k, right-to-left.

Feature set Φ	$\lvert S_{(\Phi,\phi)}\rvert$	Avg. $\lvert M_{(\Phi,\phi)}\rvert$	SD
$[\begin{smallmatrix}\textbf{nasal}\\ \text{cons}\end{smallmatrix}]$	39	32	4.3
[**high**]	39	3	0
[**low**]	39	3	0
[**tense**]	39	3	0
[**front**]	39	3	0
[**back**]	39	3	0
[**round**]	14	2	0
[**long**]	14	2	0
[**cons**]	14	2	0
[**son**]	14	2	0
[**cont**]	14	2	0
[**delrel**]	39	3	0
[**approx**]	39	3	0
[**voice**]	14	2	0
[**lab**]	39	3	0
[**labdent**]	39	3	0
[**cor**]	39	3	0
[**ant**]	39	3	0
[**distr**]	39	3	0
[**strid**]	39	3	0
[**lat**]	39	3	0
[**dor**]	39	3	0

TABLE B.5: Samples $|S_{(\Phi,\,\phi)}|$, Avg. $|M_{(\Phi,\,\phi)}|$ and standard deviation (SD) for all *sufficient* features and selected one for the *insufficient* feature for VN in English, left-to-right, learned k.

Feature set Φ	$\lvert \mathbf{S}_{(\Phi,\phi)} \rvert$	Avg. $\lvert \mathbf{M}_{(\Phi,\phi)} \rvert$	SD
$\left[\begin{smallmatrix} \textbf{nasal} \\ \text{cons} \end{smallmatrix} \right]$	39	32	3.89
[**high**]	39	3	0
[**low**]	39	3	0
[**tense**]	39	3	0
[**front**]	39	3	0
[**back**]	39	3	0
[**round**]	14	2	0
[**long**]	14	2	0
[**cons**]	14	2	0
[**son**]	14	2	0
[**cont**]	14	2	0
[**delrel**]	39	3	0
[**approx**]	39	3	0
[**voice**]	14	2	0
[**lab**]	39	3	0
[**labdent**]	39	3	0
[**cor**]	39	3	0
[**ant**]	39	3	0
[**distr**]	39	3	0
[**strid**]	39	3	0
[**lat**]	39	3	0
[**dor**]	39	3	0

TABLE B.6: Samples $\lvert S_{(\Phi,\,\phi)} \rvert$, Avg. $\lvert M_{(\Phi,\,\phi)} \rvert$ and standard deviation (SD) for all *sufficient* features and selected one for the *insufficient* feature for VN in English, right-to-left, learned k.

Feature set Φ	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD
$[\,^{\textbf{nasal}}_{\text{cons}}\,]$	39	31	3.89
[**high**]	39	24	4.52
[**low**]	39	23	7.51
[**tense**]	39	25	9.42
[**front**]	39	20	8.09
[**back**]	39	24	5.65
[**round**]	14	14	0.67
[**long**]	14	8	1.79
[**cons**]	14	9	2.02
[**son**]	14	10	1.66
[**cont**]	14	9	3.47
[**delrel**]	39	25	4.99
[**approx**]	39	26	9.29
[**voice**]	14	9	2.66
[**lab**]	39	24	4.72
[**labdent**]	39	26	4.72
[**cor**]	39	23	4.76
[**ant**]	39	24	5.80
[**distr**]	39	26	3.92
[**strid**]	39	24	8.50
[**lat**]	39	28	4.06
[**dor**]	39	25	5.38

TABLE B.7: Samples $|S_{(\Phi,\,\phi)}|$, Avg. $|M_{(\Phi,\,\phi)}|$ and standard deviation (SD) for all *sufficient* features and selected one for the *insufficient* feature for VN in English, left-to-right, set $k = 2$.

Feature set Φ	$\lvert S_{(\Phi,\phi)}\rvert$	Avg. $\lvert M_{(\Phi,\phi)}\rvert$	SD
$[\begin{smallmatrix}\textbf{nasal}\\ \text{cons}\end{smallmatrix}]$	39	29	10.10
[**high**]	39	19	8.78
[**low**]	39	12	10.94
[**tense**]	39	13	11.85
[**front**]	39	21	11.20
[**back**]	39	17	12.01
[**round**]	14	7	3.57
[**long**]	14	7	4.16
[**cons**]	14	5	2.13
[**son**]	14	9	4.06
[**cont**]	14	7	4.30
[**delrel**]	39	25	4.99
[**approx**]	39	22	11.55
[**voice**]	14	8	3.92
[**lab**]	39	16	10.77
[**labdent**]	39	25	10.96
[**cor**]	39	16	11.17
[**ant**]	39	16	12.40
[**distr**]	39	22	10.49
[**strid**]	39	17	10.70
[**lat**]	39	18	11.77
[**dor**]	39	17	11.73

TABLE B.8: Samples $|S_{(\Phi,\,\phi)}|$, Avg. $|M_{(\Phi,\,\phi)}|$ and standard deviation (SD) for all *sufficient* features and selected one for the *insufficient* feature for VN in English, right-to-left, set $k = 2$.

B.1.1 English: SEAL

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[nasal / cons]	11	20	29	29	24	37	18	24	38	36
[**high**]	2	2	2	2	2	2	2	2	2	2
[**low**]	3	3	3	3	3	3	3	3	3	3
[**tense**]	3	3	3	3	3	3	3	3	3	3
[**front**]	3	3	3	3	3	3	3	3	3	3
[**back**]	3	3	3	3	3	3	3	3	3	3
[**round**]	2	2	2	2	2	2	2	2	2	2
[**long**]	2	2	2	2	2	2	2	2	2	2
[**cons**]	2	2	2	2	2	2	2	2	2	2
[**son**]	2	2	2	2	2	2	2	2	2	2
[**cont**]	2	2	2	2	2	2	2	2	2	2
[**delrel**]	3	3	3	3	3	3	3	3	3	3
[**approx**]	3	3	3	3	3	3	3	3	3	3
[**voice**]	2	2	2	2	2	2	2	2	2	2
[**lab**]	3	3	3	3	3	3	3	3	3	3
[**labdent**]	3	3	3	3	3	3	3	3	3	3
[**cor**]	3	3	3	3	3	3	3	3	3	3
[**ant**]	3	3	3	3	3	3	3	3	3	3
[**distr**]	3	3	3	3	3	3	3	3	3	3
[**strid**]	3	3	3	3	3	3	3	3	3	3
[**lat**]	3	3	3	3	3	3	3	3	3	3
[**dor**]	3	3	3	3	3	3	3	3	3	3
$\lvert\mathbf{M}_F\rvert$	19	27	37	36	31	45	27	32	45	43

TABLE B.9: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for the *insufficient* feature, across 10 runs, for VN in English with learned k, left-to-right, with SEAL.

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[$\begin{smallmatrix}\textbf{nasal}\\ \textbf{cons}\end{smallmatrix}$]	4	8	9	9	7	6	4	4	7	13
[**high**]	2	2	2	2	2	2	2	2	2	2
[**low**]	3	3	3	3	3	3	3	3	3	3
[**tense**]	3	3	3	3	3	3	3	3	3	3
[**front**]	3	3	3	3	3	3	3	3	3	3
[**back**]	3	3	3	3	3	3	3	3	3	3
[**round**]	2	2	2	2	2	2	2	2	2	2
[**long**]	2	2	2	2	2	2	2	2	2	2
[**cons**]	2	2	2	2	2	2	2	2	2	2
[**son**]	2	2	2	2	2	2	2	2	2	2
[**cont**]	2	2	2	2	2	2	2	2	2	2
[**delrel**]	3	3	3	3	3	3	3	3	3	3
[**approx**]	3	3	3	3	3	3	3	3	3	3
[**voice**]	2	2	2	2	2	2	2	2	2	2
[**lab**]	3	3	3	3	3	3	3	3	3	3
[**labdent**]	3	3	3	3	3	3	3	3	3	3
[**cor**]	3	3	3	3	3	3	3	3	3	3
[**ant**]	3	3	3	3	3	3	3	3	3	3
[**distr**]	3	3	3	3	3	3	3	3	3	3
[**strid**]	3	3	3	3	3	3	3	3	3	3
[**lat**]	3	3	3	3	3	3	3	3	3	3
[**dor**]	3	3	3	3	3	3	3	3	3	3
$\lvert\mathbf{M}_F\rvert$	11	16	16	17	14	13	13	11	15	20

TABLE B.10: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for the *insufficient* feature, across 10 runs, for VN in English with learned k, right-to-left, with SEAL.

Feature set Φ	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD
$[\begin{smallmatrix}\textbf{nasal}\\ \text{cons}\end{smallmatrix}]$	39	27	8.90
[**high**]	39	3	0
[**low**]	39	3	0
[**tense**]	39	3	0
[**front**]	39	3	0
[**back**]	39	3	0
[**round**]	14	2	0
[**long**]	14	2	0
[**cons**]	14	2	0
[**son**]	14	2	0
[**cont**]	14	2	0
[**delrel**]	39	3	0
[**approx**]	39	3	0
[**voice**]	14	2	0
[**lab**]	39	3	0
[**labdent**]	39	3	0
[**cor**]	39	3	0
[**ant**]	39	3	0
[**distr**]	39	3	0
[**strid**]	39	3	0
[**lat**]	39	3	0
[**dor**]	39	3	0

TABLE B.11: Samples $|S_{(\Phi,\,\phi)}|$, Avg. $|M_{(\Phi,\,\phi)}|$ and standard deviation (SD) for all *sufficient* features and selected one for the *insufficient* feature for VN in English, left-to-right, learned k, with SEAL.

Feature set Φ	$\lvert \mathbf{S}_{(\Phi,\phi)} \rvert$	Avg. $\lvert \mathbf{M}_{(\Phi,\phi)} \rvert$	SD
$[\begin{smallmatrix}\textbf{nasal}\\ \text{cons}\end{smallmatrix}]$	39	7	2.85
[**high**]	39	3	0
[**low**]	39	3	0
[**tense**]	39	3	0
[**front**]	39	3	0
[**back**]	39	3	0
[**round**]	14	2	0
[**long**]	14	2	0
[**cons**]	14	2	0
[**son**]	14	2	0
[**cont**]	14	2	0
[**delrel**]	39	3	0
[**approx**]	39	3	0
[**voice**]	14	2	0
[**lab**]	39	3	0
[**labdent**]	39	3	0
[**cor**]	39	3	0
[**ant**]	39	3	0
[**distr**]	39	3	0
[**strid**]	39	3	0
[**lat**]	39	3	0
[**dor**]	39	3	0

TABLE B.12: Samples $\lvert S_{(\Phi,\,\phi)} \rvert$, Avg. $\lvert M_{(\Phi,\,\phi)} \rvert$ and standard deviation (SD) for all *sufficient* features and selected one for the *insufficient* feature for VN in English, right-to-left, learned k, with SEAL.

B.2 Yawelmani

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**high**]	106	151	127	146	140	184	119	174	145	187
[**low**]	38	33	27	37	31	38	40	29	29	17
[**tense**]	143	199	165	172	183	179	96	120	149	185
[**front**]	113	130	194	141	116	159	161	107	155	188
[**back**]	169	174	157	119	180	187	200	166	140	181
[**round**]	40	23	38	31	24	30	35	16	34	28
[**long**]	354	341	335	298	236	224	358	351	332	344
[**cons**]	33	32	37	33	46	42	35	40	27	37
[**son**]	52	36	42	27	44	32	46	38	31	44
[**cont**]	38	39	22	36	33	39	25	38	37	29
[**delrel**]	40	28	24	37	29	32	26	46	29	44
[**nasal**]	41	40	23	24	28	32	46	37	33	28
[**approx**]	40	32	44	52	28	41	44	43	39	42
[**voice**]	24	37	31	45	26	22	51	20	35	36
[**lab**]	32	39	32	36	32	37	38	28	41	31
[**labdent**]	31	31	37	26	41	32	33	38	31	29
[**cor**]	153	132	117	128	179	159	97	151	154	124
[**ant**]	128	119	132	152	104	158	148	124	131	166
[**distr**]	27	45	35	22	25	33	27	44	34	24
[**strid**]	26	21	35	23	33	31	36	41	35	23
[**lat**]	23	32	35	26	25	33	39	33	33	31
[**dor**]	142	143	181	189	169	168	164	168	199	167
[**constrgl**]	128	140	137	217	132	145	159	195	133	200
[**spreadgl**]	30	35	49	25	37	42	39	25	32	38
$\lvert\mathbf{M}_F\rvert$	920	943	942	967	868	910	972	994	952	1052

TABLE B.13: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features, across 10 runs, for VS in Yawelmani, left-to-right, $k = 3$.

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**high**]	185	129	116	132	163	159	157	171	144	117
[**low**]	25	43	29	27	30	23	3	46	37	39
[**tense**]	184	139	176	123	170	186	143	166	201	133
[**front**]	152	117	129	180	126	168	135	189	109	123
[**back**]	86	129	183	172	159	167	158	132	161	159
[**round**]	38	25	46	35	31	48	31	29	36	34
[**long**]	162	189	131	166	169	194	168	176	156	175
[**cons**]	34	22	44	26	31	3	43	32	31	29
[**son**]	31	32	28	32	38	29	34	30	37	40
[**cont**]	44	6	45	49	38	31	44	43	25	29
[**delrel**]	47	40	34	49	41	31	37	48	34	30
[**nasal**]	28	51	23	38	38	3	43	32	31	40
[**approx**]	33	37	41	33	19	41	30	29	31	20
[**voice**]	37	37	31	34	30	35	36	28	32	20
[**lab**]	37	45	34	3	36	33	32	27	37	3
[**labdent**]	32	34	30	38	17	30	34	33	27	45
[**cor**]	114	155	247	181	236	156	137	183	211	121
[**ant**]	4	197	183	275	243	246	226	166	210	202
[**distr**]	40	32	28	34	35	38	30	48	45	3
[**strid**]	33	28	39	32	20	23	32	3	32	34
[**lat**]	35	18	3	3	20	25	31	23	31	28
[**dor**]	265	211	197	208	235	251	137	119	135	169
[**constrgl**]	154	148	238	167	154	226	227	128	4	138
[**spreadgl**]	48	29	3	3	42	34	41	30	30	34
$\lvert\mathbf{M}_F\rvert$	787	846	851	901	936	934	834	836	795	785

TABLE B.14: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for the *insufficient* feature, across 10 runs, for VS in Yawelamani with set $k = 3$, right-to-left.

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**high**]	3	3	3	3	3	3	3	3	3	3
[**low**]	2	2	2	2	2	2	2	2	2	2
[**tense**]	3	3	3	3	3	3	3	3	3	3
[**front**]	3	3	3	3	3	3	3	3	3	3
[**back**]	3	3	3	3	3	3	3	3	3	3
[**round**]	2	2	2	2	2	2	2	2	2	2
[**long**]	289	216	337	186	264	333	162	303	224	192
[**cons**]	2	2	2	2	2	2	2	2	2	2
[**son**]	2	2	2	2	2	2	2	2	2	2
[**cont**]	2	2	2	2	2	2	2	2	2	2
[**delrel**]	2	2	2	2	2	2	2	2	2	2
[**nasal**]	2	2	2	2	2	2	2	2	2	2
[**approx**]	2	2	2	2	2	2	2	2	2	2
[**voice**]	2	2	2	2	2	2	2	2	2	2
[**lab**]	2	2	2	2	2	2	2	2	2	2
[**labdent**]	2	2	2	2	2	2	2	2	2	2
[**cor**]	3	3	3	3	3	3	3	3	3	3
[**ant**]	3	3	3	3	3	3	3	3	3	3
[**distr**]	2	2	2	2	2	2	2	2	2	2
[**strid**]	2	2	2	2	2	2	2	2	2	2
[**lat**]	2	2	2	2	2	2	2	2	2	2
[**dor**]	3	3	3	3	3	3	3	3	3	3
[**constrgl**]	3	3	3	3	3	3	3	3	3	3
[**spreadgl**]	2	2	2	2	2	2	2	2	2	2
$\lvert\mathbf{M}_F\rvert$	297	223	343	195	274	340	168	310	234	200

TABLE B.15: Cardinalities $|M_{(\Phi,\phi)}|$ and $|M_F|$ for all *sufficient* features, across 10 runs, for VS in Yawelmani, left-to-right, learned k.

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**high**]	3	3	3	3	3	3	3	3	3	3
[**low**]	2	2	2	2	2	2	2	2	2	2
[**tense**]	3	3	3	3	3	3	3	3	3	3
[**front**]	3	3	3	3	3	3	3	3	3	3
[**back**]	3	3	3	3	3	3	3	3	3	3
[**round**]	2	2	2	2	2	2	2	2	2	2
[**long**]	127	188	149	137	147	156	152	157	193	140
[**cons**]	2	2	2	2	2	2	2	2	2	2
[**son**]	2	2	2	2	2	2	2	2	2	2
[**cont**]	2	2	2	2	2	2	2	2	2	2
[**delrel**]	2	2	2	2	2	2	2	2	2	2
[**nasal**]	2	2	2	2	2	2	2	2	2	2
[**approx**]	2	2	2	2	2	2	2	2	2	2
[**voice**]	2	2	2	2	2	2	2	2	2	2
[**lab**]	2	2	2	2	2	2	2	2	2	2
[**labdent**]	2	2	2	2	2	2	2	2	2	2
[**cor**]	3	3	3	3	3	3	3	3	3	3
[**ant**]	3	3	3	3	3	3	3	3	3	3
[**distr**]	2	2	2	2	2	2	2	2	2	2
[**strid**]	2	2	2	2	2	2	2	2	2	2
[**lat**]	2	2	2	2	2	2	2	2	2	2
[**dor**]	3	3	3	3	3	3	3	3	3	3
[**constrgl**]	3	3	3	3	3	3	3	3	3	3
[**spreadgl**]	2	2	2	2	2	2	2	2	2	2
$\lvert\mathbf{M}_F\rvert$	137	198	155	146	157	168	161	167	203	151

TABLE B.16: Cardinalities $|M_{(\Phi,\phi)}|$ and $|M_F|$ for all *sufficient* features, across 10 runs, for VS in Yawelmani, right-to-left, learned k.

Feature set Φ	$\lvert S_{(\Phi,\phi)}\rvert$	Avg. $\lvert M_{(\Phi,\phi)}\rvert$	SD
[**high**]	363	3	0
[**low**]	62	2	0
[**tense**]	363	3	0
[**front**]	363	3	0
[**back**]	363	3	0
[**round**]	62	2	0
[**long**]	363	251	63.26
[**cons**]	62	2	0
[**son**]	62	2	0
[**cont**]	62	2	0
[**delrel**]	62	2	0
[**approx**]	62	2	0
[**nasal**]	54	2	0
[**voice**]	54	2	0
[**lab**]	62	2	0
[**labdent**]	62	2	0
[**cor**]	336	3	0
[**ant**]	309	3	0
[**distr**]	62	2	0
[**strid**]	62	2	0
[**lat**]	62	2	0
[**dor**]	336	3	0
[**constrgl**]	336	3	0
[**spreadgl**]	62	2	0

TABLE B.17: Samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for all *sufficient* features for VS in Yawelmani, left-to-right, learned k.

Feature set Φ	$\lvert S_{(\Phi,\phi)}\rvert$	Avg. $\lvert M_{(\Phi,\phi)}\rvert$	SD
[**high**]	363	3	0
[**low**]	62	2	0
[**tense**]	363	3	0
[**front**]	363	3	0
[**back**]	363	3	0
[**round**]	62	2	0
[**long**]	363	155	21.02
[**cons**]	62	2	0
[**son**]	62	2	0
[**cont**]	62	2	0
[**delrel**]	62	2	0
[**approx**]	62	2	0
[**nasal**]	54	2	0
[**voice**]	54	2	0
[**lab**]	62	2	0
[**labdent**]	62	2	0
[**cor**]	336	3	0
[**ant**]	309	3	0
[**distr**]	62	2	0
[**strid**]	62	2	0
[**lat**]	62	2	0
[**dor**]	336	3	0
[**constrgl**]	336	3	0
[**spreadgl**]	62	2	0

TABLE B.18: Samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for all *sufficient* features for VS in Yawelmani, right-to-left, learned k.

Feature set Φ	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD
[**high**]	363	148	27.13
[**low**]	62	32	6.92
[**tense**]	363	159	32.15
[**front**]	363	146	30.43
[**back**]	363	167	23.66
[**round**]	62	30	7.39
[**long**]	363	317	2.63
[**cons**]	62	36	49.02
[**son**]	62	39	5.47
[**cont**]	62	34	7.77
[**delrel**]	62	34	6.19
[**approx**]	62	33	7.75
[**nasal**]	54	41	6.68
[**voice**]	54	33	10.11
[**lab**]	62	35	4.17
[**labdent**]	62	33	4.51
[**cor**]	336	139	24.05
[**ant**]	309	136	19.28
[**distr**]	62	32	8.09
[**strid**]	62	30	6.75
[**lat**]	62	31	4.92
[**dor**]	336	169	17.89
[**constrgl**]	336	157	32.91
[**spreadgl**]	62	35	7.51

TABLE B.19: Samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for all *sufficient* features for VS in Yawelmani, left-to-right, set $k = 3$.

Feature set Φ	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD
[**high**]	363	147	23.42
[**low**]	62	30	12.33
[**tense**]	363	162	26.05
[**front**]	363	143	27.83
[**back**]	363	151	28.02
[**round**]	62	35	7.21
[**long**]	363	167	17.58
[**cons**]	62	30	11.52
[**son**]	62	33	4.04
[**cont**]	62	35	13.01
[**delrel**]	62	39	7.06
[**approx**]	62	31	7.55
[**nasal**]	54	33	13.13
[**voice**]	54	32	5.21
[**lab**]	62	29	14.29
[**labdent**]	62	32	7.24
[**cor**]	336	174	46.11
[**ant**]	309	195	74.54
[**distr**]	62	33	12.39
[**strid**]	62	28	10.23
[**lat**]	62	22	11.15
[**dor**]	336	193	51.02
[**constrgl**]	336	158	67.51
[**spreadgl**]	62	29	15.19

TABLE B.20: Samples $\lvert\mathrm{S}_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert\mathrm{M}_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for all *sufficient* features for VS in Yawelmani, right-to-left, set $k = 3$.

B.2.1 Yawelmani: SEAL

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**high**]	3	3	3	3	3	3	3	3	3	3
[**low**]	2	2	2	2	2	2	2	2	2	2
[**tense**]	3	3	3	3	3	3	3	3	3	3
[**front**]	3	3	3	3	3	3	3	3	3	3
[**back**]	3	3	3	3	3	3	3	3	3	3
[**round**]	2	2	2	2	2	2	2	2	2	2
[**long**]	276	313	288	358	357	321	317	353	235	272
[**cons**]	2	2	2	2	2	2	2	2	2	2
[**son**]	2	2	2	2	2	2	2	2	2	2
[**cont**]	2	2	2	2	2	2	2	2	2	2
[**delrel**]	2	2	2	2	2	2	2	2	2	2
[**nasal**]	2	2	2	2	2	2	2	2	2	2
[**approx**]	2	2	2	2	2	2	2	2	2	2
[**voice**]	2	2	2	2	2	2	2	2	2	2
[**lab**]	2	2	2	2	2	2	2	2	2	2
[**labdent**]	2	2	2	2	2	2	2	2	2	2
[**cor**]	3	3	3	3	3	3	3	3	3	3
[**ant**]	3	3	3	3	3	3	3	3	3	3
[**distr**]	2	2	2	2	2	2	2	2	2	2
[**strid**]	2	2	2	2	2	2	2	2	2	2
[**lat**]	2	2	2	2	2	2	2	2	2	2
[**dor**]	3	3	3	3	3	3	3	3	3	3
[**constrgl**]	3	3	3	3	3	3	3	3	3	3
[**spreadgl**]	2	2	2	2	2	2	2	2	2	2
$\lvert\mathbf{M}_F\rvert$	279	316	291	361	360	324	320	356	238	275

TABLE B.21: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features, across 10 runs, for VS in Yawelmani, left-to-right, learned k, with SEAL.

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**high**]	3	3	3	3	3	3	3	3	3	3
[**low**]	2	2	2	2	2	2	2	2	2	2
[**tense**]	3	3	3	3	3	3	3	3	3	3
[**front**]	3	3	3	3	3	3	3	3	3	3
[**back**]	3	3	3	3	3	3	3	3	3	3
[**round**]	2	2	2	2	2	2	2	2	2	2
[**long**]	26	28	11	6	10	14	6	31	34	5
[**cons**]	2	2	2	2	2	2	2	2	2	2
[**son**]	2	2	2	2	2	2	2	2	2	2
[**cont**]	2	2	2	2	2	2	2	2	2	2
[**delrel**]	2	2	2	2	2	2	2	2	2	2
[**nasal**]	2	2	2	2	2	2	2	2	2	2
[**approx**]	2	2	2	2	2	2	2	2	2	2
[**voice**]	2	2	2	2	2	2	2	2	2	2
[**lab**]	2	2	2	2	2	2	2	2	2	2
[**labdent**]	2	2	2	2	2	2	2	2	2	2
[**cor**]	3	3	3	3	3	3	3	3	3	3
[**ant**]	3	3	3	3	3	3	3	3	3	3
[**distr**]	2	2	2	2	2	2	2	2	2	2
[**strid**]	2	2	2	2	2	2	2	2	2	2
[**lat**]	2	2	2	2	2	2	2	2	2	2
[**dor**]	3	3	3	3	3	3	3	3	3	3
[**constrgl**]	3	3	3	3	3	3	3	3	3	3
[**spreadgl**]	2	2	2	2	2	2	2	2	2	2
$\lvert\mathbf{M}_F\rvert$	29	31	14	9	13	17	9	34	37	8

TABLE B.22: Cardinalities $|M_{(\Phi,\phi)}|$ and $|M_F|$ for all *sufficient* features, across 10 runs, for VS in Yawelmani, right-to-left, learned k, with SEAL.

Feature set Φ	$\lvert S_{(\Phi,\phi)}\rvert$	Avg. $\lvert M_{(\Phi,\phi)}\rvert$	SD
[**high**]	363	3	0
[**low**]	62	2	0
[**tense**]	363	3	0
[**front**]	363	3	0
[**back**]	363	3	0
[**round**]	62	2	0
[**long**]	363	309	41.15
[**cons**]	62	2	0
[**son**]	62	2	0
[**cont**]	62	2	0
[**delrel**]	62	2	0
[**approx**]	62	2	0
[**nasal**]	54	2	0
[**voice**]	54	2	0
[**lab**]	62	2	0
[**labdent**]	62	2	0
[**cor**]	336	3	0
[**ant**]	309	3	0
[**distr**]	62	2	0
[**strid**]	62	2	0
[**lat**]	62	2	0
[**dor**]	336	3	0
[**constrgl**]	336	3	0
[**spreadgl**]	62	2	0

TABLE B.23: Samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for all *sufficient* features for VS in Yawelmani, left-to-right, learned k, with SEAL.

Feature set Φ	$\lvert S_{(\Phi,\phi)}\rvert$	Avg. $\lvert M_{(\Phi,\phi)}\rvert$	SD
[**high**]	363	3	0
[**low**]	62	2	0
[**tense**]	363	3	0
[**front**]	363	3	0
[**back**]	363	3	0
[**round**]	62	2	0
[**long**]	363	17	11.39
[**cons**]	62	2	0
[**son**]	62	2	0
[**cont**]	62	2	0
[**delrel**]	62	2	0
[**approx**]	62	2	0
[**nasal**]	54	2	0
[**voice**]	54	2	0
[**lab**]	62	2	0
[**labdent**]	62	2	0
[**cor**]	336	3	0
[**ant**]	309	3	0
[**distr**]	62	2	0
[**strid**]	62	2	0
[**lat**]	62	2	0
[**dor**]	336	3	0
[**constrgl**]	336	3	0
[**spreadgl**]	62	2	0

TABLE B.24: Samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for all *sufficient* features for VS in Yawelmani, right-to-left, learned k, with SEAL.

B.3 Chukchi

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**high**]	193	146	174	176	125	170	228	154	137	144
[**low**]	30	40	44	32	45	39	44	37	36	32
[**tense**]	32	22	39	32	32	26	29	32	41	36
[**front**]	123	172	191	183	144	206	147	193	186	114
[**back**]	190	190	188	174	160	204	134	152	194	157
[**round** / high]	256	186	215	222	239	254	209	170	260	236
[**long**]	28	34	42	29	30	36	22	32	37	34
[**cons**]	37	33	31	28	31	32	32	32	19	36
[**son** / low]	223	259	203	253	273	283	283	271	267	287
[**cont** / tense]	277	315	317	323	317	293	313	307	275	323
[**delrel** / long]	1023	933	873	923	863	853	913	963	893	1003
[**approx**]	26	32	38	45	28	27	38	38	39	32
[**nasal** / son]	277	298	283	267	297	331	281	285	293	285
[**voice** / cons]	277	319	259	311	295	317	323	321	311	313
[**lab**]	315	311	323	311	321	317	297	305	303	323
[**labdent**]	39	44	31	29	35	33	28	26	36	36
[**cor**]	315	285	313	267	299	311	321	307	315	321
[**ant** / labdent]	873	873	893	963	993	923	1053	883	873	893
[**distr** / cor]	813	813	813	784	784	784	813	813	784	813
[**strid** / dor]	963	893	1013	1083	953	933	963	953	893	903
[**lat**]	279	249	257	267	287	287	285	275	275	281
[**dor**]	34	37	26	32	30	40	31	37	30	27
$\lvert\mathbf{M}_F\rvert$	2,380	2,293	2,320	2,373	2,301	2,308	2,372	2,365	2,238	2,356

TABLE B.25: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for each *insufficient* feature, across 10 runs, for FE in Chukchi, left-to-right, set $k = 3$.

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**high**]	135	193	235	279	256	154	209	143	163	150
[**low**]	21	45	36	38	41	39	40	43	37	45
[**tense**]	46	37	25	33	45	38	17	35	48	36
[**front**]	154	179	190	197	222	214	218	148	177	209
[**back**]	210	232	133	130	176	233	171	214	138	110
[**round** high]	246	208	195	180	258	199	244	297	237	207
[**long**]	35	38	44	38	45	42	39	34	39	42
[**cons**]	38	37	38	36	41	39	43	39	45	34
[**son** low]	311	222	207	247	314	236	203	278	225	199
[**cont** tense]	260	258	180	190	252	205	196	244	198	202
[**delrel** long]	923	963	1,073	1,013	953	983	993	983	943	993
[**approx**]	40	40	46	36	47	39	39	37	36	34
[**nasal** son]	191	282	260	232	200	252	273	152	262	216
[**voice** cons]	193	202	167	244	216	214	252	256	292	240
[**lab**]	128	172	128	190	138	152	160	135	128	188
[**labdent**]	40	39	48	42	46	35	31	46	39	40
[**cor**]	126	159	138	119	132	135	104	250	160	156
[**ant** labdent]	943	953	1,123	1,093	983	923	883	1,133	893	1,003
[**distr** cor]	983	973	883	893	933	863	943	993	863	843
[**strid** dor]	893	983	973	993	923	943	963	1,063	933	973
[**lat**]	182	198	171	234	242	155	133	172	154	112
[**dor**]	42	42	42	36	50	31	44	47	48	39
$\lvert\mathbf{M}_F\rvert$	2,315	2,388	2,411	2,380	2,340	2,295	2,334	2,416	2,311	2,332

TABLE B.26: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for each *insufficient* feature, across 10 runs, for FE in Chukchi, right-to-left, set $k = 3$.

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**high**]	143	162	197	176	216	152	151	156	139	161
[**low**]	38	28	31	34	33	42	36	33	28	24
[**tense**]	48	28	28	30	39	32	32	32	38	40
[**front**]	122	126	157	141	175	200	143	138	133	123
[**back**]	126	189	166	193	171	198	125	118	125	126
[**round** / high]	237	263	254	219	186	232	146	217	278	216
[**long**]	27	42	39	33	31	33	35	38	41	50
[**cons**]	30	35	39	31	32	38	33	30	42	41
[**son** / low]	275	279	287	275	287	273	277	287	277	281
[**cont** / tense]	321	277	307	299	319	309	295	285	297	293
[**delrel** / long]	953	1227	973	883	933	903	903	973	853	963
[**approx**]	28	31	38	41	33	34	36	26	46	32
[**nasal** / son]	269	291	329	281	314	331	298	247	282	325
[**voice** / cons]	303	297	323	283	301	307	305	317	287	323
[**lab**]	301	319	281	257	301	323	323	315	295	305
[**labdent**]	42	34	32	47	30	32	32	35	35	35
[**cor**]	291	275	303	293	321	307	321	285	261	305
[**ant** / labdent]	953	1217	853	943	943	863	913	893	923	963
[**distr** / cor]	953	1236	743	743	743	743	743	743	743	743
[**strid** / dor]	953	1236	903	903	983	933	953	883	943	883
[**lat**]	279	285	237	231	237	287	253	285	251	279
[**dor**]	32	33	30	36	30	30	34	38	30	33
$\lvert\mathbf{M}_F\rvert$	2,382	2,599	2,297	2,291	2,370	2,270	2,271	2,276	2,275	2,350

TABLE B.27: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for each *insufficient* feature, across 10 runs, for FE in Chukchi, left-to-right, learned k.

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**high**]	125	211	171	190	110	214	215	148	188	153
[**low**]	40	44	44	34	36	34	36	25	35	37
[**tense**]	40	43	41	39	47	40	51	36	43	45
[**front**]	184	171	207	186	233	170	205	229	130	217
[**back**]	232	135	172	215	174	225	186	275	213	135
[**round** high]	256	257	167	252	230	163	174	186	225	249
[**long**]	31	37	40	37	36	38	41	38	38	40
[**cons**]	27	34	40	45	36	36	49	41	37	46
[**son** low]	204	208	258	250	176	208	298	308	191	256
[**cont** tense]	201	199	270	221	230	208	190	228	236	280
[**delrel** long]	1,063	1,033	1,183	1,073	1,123	1,033	953	963	893	1,073
[**approx**]	48	44	21	38	45	37	42	43	39	36
[**nasal** son]	225	208	227	230	190	311	286	302	192	170
[**voice** cons]	320	278	213	276	212	224	202	300	184	206
[**lab**]	102	142	228	132	150	170	162	148	162	116
[**labdent**]	46	39	46	29	33	44	42	41	49	43
[**cor**]	206	145	172	108	186	166	164	115	186	147
[**ant** labdent]	1,103	963	1,023	973	913	1,033	1,163	903	883	1,133
[**distr** cor]	1,083	923	873	973	963	1,013	973	933	843	883
[**strid** dor]	1,063	1,123	1,113	873	993	963	1,033	923	953	873
[**lat**]	126	172	116	140	226	176	266	166	266	219
[**dor**]	43	39	44	35	39	37	42	35	53	38
$\lvert\mathbf{M}_F\rvert$	2,481	2,426	2,524	2,419	2,451	2,391	2,364	2,341	2,264	2,462

TABLE B.28: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for each *insufficient* feature, across 10 runs, for FE in Chukchi, right-to-left, learned k.

Feature set Φ	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD
[**high**]	309	165	24.44
[**low**]	62	33	5.27
[**tense**]	62	35	6.40
[**front**]	309	146	25.05
[**back**]	309	154	32.67
[**round** high]	309	225	38.34
[**long**]	62	37	6.56
[**cons**]	62	35	4.58
[**son** low]	336	280	5.53
[**cont** tense]	363	300	14.02
[**delrel** long]	1,236	956	103.37
[**approx**]	62	35	6.00
[**nasal** son]	336	297	28.00
[**voice** cons]	363	305	13.69
[**lab**]	363	302	20.73
[**labdent**]	62	35	5.21
[**cor**]	363	296	19.21
[**ant** labdent]	1,236	946	102.07
[**distr** cor]	1,236	813	162.52
[**strid** dor]	1,236	957	103.34
[**lat**]	336	262	22.78
[**dor**]	62	33	2.80

TABLE B.29: Samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for all *sufficient* features and selected one for each *insufficient* feature for FE in Chukchi, left-to-right, learned k.

Feature set Φ	$\lvert S_{(\Phi,\phi)}\rvert$	Avg. $\lvert M_{(\Phi,\phi)}\rvert$	SD
[**high**]	309	173	37.57
[**low**]	62	37	5.50
[**tense**]	62	43	4.33
[**front**]	309	193	31.49
[**back**]	309	196	44.26
[**round** / high]	309	216	39.18
[**long**]	62	38	2.80
[**cons**]	62	39	6.51
[**son** / low]	336	236	45.12
[**cont** / tense]	363	226	29.76
[**delrel** / long]	1,236	1,039	85.14
[**approx**]	62	39	7.48
[**nasal** / son]	336	234	49.30
[**voice** / cons]	363	242	47.40
[**lab**]	363	151	34.33
[**labdent**]	62	41	6.14
[**cor**]	363	160	31.24
[**ant** / labdent]	1,236	1,009	99.13
[**distr** / cor]	1,236	946	71.19
[**strid** / dor]	1,236	991	90.65
[**lat**]	336	187	54.55
[**dor**]	62	41	5.38

TABLE B.30: Samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for all *sufficient* features and selected one for each *insufficient* feature for FE in Chukchi, left-to-right, learned k.

Feature set Φ	$\lvert S_{(\Phi,\phi)}\rvert$	Avg. $\lvert M_{(\Phi,\phi)}\rvert$	SD
[**high**]	309	165	30.34
[**low**]	62	38	5.45
[**tense**]	62	32	5.69
[**front**]	309	166	31.76
[**back**]	309	174	22.55
[**round** high]	309	225	30.28
[**long**]	62	32	5.54
[**cons**]	62	31	4.95
[**son** low]	336	260	12.77
[**cont** tense]	363	306	18.02
[**delrel** long]	1,236	924	57.82
[**approx**]	62	34	6.24
[**nasal** son]	336	290	17.26
[**voice** cons]	363	305	21.22
[**lab**]	363	313	8.88
[**labdent**]	62	34	5.46
[**cor**]	363	305	17.30
[**ant** labdent]	1,236	922	61.73
[**distr** cor]	1,236	801	14.98
[**strid** dor]	1,236	955	58.46
[**lat**]	336	274	12.90
[**dor**]	62	32	4.55

TABLE B.31: Samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for all *sufficient* features and selected one for each *insufficient* feature for FE in Chukchi, left-to-right, set $k = 3$.

Feature set Φ	$\lvert S_{(\Phi,\phi)}\rvert$	Avg. $\lvert M_{(\Phi,\phi)}\rvert$	SD
[**high**]	309	192	51.10
[**low**]	62	39	6.90
[**tense**]	62	36	9.56
[**front**]	309	191	26.11
[**back**]	309	175	45.65
[**round** high]	309	227	35.57
[**long**]	62	40	3.63
[**cons**]	62	39	3.27
[**son** low]	336	244	19.54
[**cont** tense]	363	219	31.15
[**delrel** long]	1,236	982	41.75
[**approx**]	62	39	4.22
[**nasal** son]	336	232	41.67
[**voice** cons]	363	228	36.23
[**lab**]	363	152	24.43
[**labdent**]	62	41	5.21
[**cor**]	363	148	40.16
[**ant** labdent]	1,236	993	92.98
[**distr** cor]	1,236	917	55.02
[**strid** dor]	1,236	964	46.30
[**lat**]	336	175	41.02
[**dor**]	62	42	5.72

TABLE B.32: Samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for all *sufficient* features and selected one for each *insufficient* feature for FE in Chukchi, right-to-left, set $k = 3$.

B.3.1 Chukchi: SEAL

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**high**]	42	72	96	68	113	128	119	245	229	230
[**low**]	55	44	36	33	41	48	41	56	40	17
[**tense**]	22	55	15	34	62	11	22	16	51	12
[**front**]	73	284	103	293	103	112	101	97	274	305
[**back**]	73	173	166	158	111	136	113	173	221	213
[$\substack{\textbf{round}\\ \text{high}}$]	13	18	13	4	11	8	12	18	6	9
[**long**]	36	36	29	37	37	25	13	41	32	59
[**cons**]	49	54	30	62	52	35	59	37	11	11
[$\substack{\textbf{son}\\ \text{low}}$]	292	317	211	334	266	324	205	308	331	150
[$\substack{\textbf{cont}\\ \text{tense}}$]	270	326	234	65	99	358	322	317	300	306
[$\substack{\textbf{delrel}\\ \text{long}}$]	1,184	1,224	1,184	1,204	1,064	1,024	1,144	924	1,024	1,224
[**approx**]	10	47	26	56	13	21	33	47	38	47
[$\substack{\textbf{nasal}\\ \text{son}}$]	155	327	295	202	296	273	321	203	166	201
[$\substack{\textbf{voice}\\ \text{cons}}$]	307	208	205	188	303	322	275	74	304	297
[**lab**]	331	272	202	359	302	358	342	349	343	260
[**labdent**]	18	47	50	12	47	37	22	13	16	26
[**cor**]	362	238	307	296	349	263	351	350	325	303
[$\substack{\textbf{ant}\\ \text{labdent}}$]	1,204	1,084	1,184	1,144	1,184	724	1,124	1,204	1,144	1,204
[$\substack{\textbf{distr}\\ \text{cor}}$]	1,144	1,204	1,184	664	1,124	684	1,164	1,224	1,144	1,104
[$\substack{\textbf{strid}\\ \text{dor}}$]	1,124	1,164	1,224	1,184	1,024	1,164	924	844	804	1,044
[**lat**]	218	293	262	257	182	294	316	197	225	227
[**dor**]	32	9	33	18	52	17	21	38	37	32
$\lvert\mathbf{M}_F\rvert$	2,443	2,612	2,461	2,502	2,344	2,452	2,460	2,297	2,454	2,641

TABLE B.33: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for each *insufficient* feature, across 10 runs, for FE in Chukchi, left-to-right, learned k, with SEAL.

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**high**]	38	33	41	52	83	56	53	49	139	37
[**low**]	10	10	16	11	10	10	14	10	9	10
[**tense**]	15	22	16	10	10	7	9	12	10	10
[**front**]	38	44	39	54	80	43	55	35	30	88
[**back**]	32	60	46	69	39	48	47	56	93	49
[**round** high]	53	64	71	48	45	49	58	44	41	91
[**long**]	14	18	12	11	21	16	14	13	18	10
[**cons**]	11	13	9	13	15	21	18	10	7	19
[**son** low]	36	41	53	49	35	47	53	72	56	60
[**cont** tense]	43	35	41	37	44	76	53	64	32	37
[**delrel** long]	139	93	196	130	147	85	147	75	143	214
[**approx**]	15	14	11	10	20	8	14	13	14	14
[**nasal** son]	59	42	56	67	36	30	34	39	76	60
[**voice** cons]	34	32	34	39	49	44	58	50	43	56
[**lab**]	38	42	45	43	53	36	33	59	41	36
[**labdent**]	18	9	15	16	10	7	12	13	14	11
[**cor**]	41	75	51	42	47	46	65	49	58	45
[**ant** labdent]	128	86	297	89	117	221	124	122	137	108
[**distr** cor]	106	136	143	164	108	105	134	86	92	200
[**strid** dor]	102	88	93	119	205	109	87	124	116	120
[**lat**]	35	42	53	39	64	31	38	47	56	39
[**dor**]	14	15	11	11	9	18	10	20	10	12
$\lvert\mathbf{M}_F\rvert$	592	566	804	669	729	665	647	590	689	771

TABLE B.34: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for each *insufficient* feature, across 10 runs, for FE in Chukchi, right-to-left, learned k, with SEAL.

Feature set Φ	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD
[**high**]	309	132	74.06
[**low**]	62	41	11.28
[**tense**]	62	30	19.27
[**front**]	309	175	99.3
[**back**]	309	154	46.27
[**round** / high]	309	11	4.64
[**long**]	62	35	11.78
[**cons**]	62	40	18.51
[**son** / low]	336	274	63.99
[**cont** / tense]	363	260	99.72
[**delrel** / long]	1,236	1,120	104.05
[**approx**]	62	34	15.84
[**nasal** / son]	336	244	65.20
[**voice** / cons]	363	248	78.58
[**lab**]	363	312	52.09
[**labdent**]	62	29	15.08
[**cor**]	363	314	41.07
[**ant** / labdent]	1,236	1,120	144.78
[**distr** / cor]	1,236	1,064	208.70
[**strid** / dor]	1,236	1,050	148.79
[**lat**]	336	247	44.55
[**dor**]	62	29	12.64

TABLE B.35: Samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for all *sufficient* features and selected one for each *insufficient* feature for FE in Chukchi, left-to-right, learned k, with SEAL.

Feature set Φ	$\lvert S_{(\Phi,\phi)}\rvert$	Avg. $\lvert M_{(\Phi,\phi)}\rvert$	SD
[**high**]	309	58	31.78
[**low**]	62	11	2.21
[**tense**]	62	12	4.55
[**front**]	309	51	19.31
[**back**]	309	54	16.34
[**round** high]	309	56	15.38
[**long**]	62	15	3.70
[**cons**]	62	14	4.60
[**son** low]	336	50	10.76
[**cont** tense]	363	46	10.14
[**delrel** long]	1,236	137	47.67
[**approx**]	62	13	3.23
[**nasal** son]	336	50	16.31
[**voice** cons]	363	44	9.26
[**lab**]	363	43	8.39
[**labdent**]	62	13	3.38
[**cor**]	363	52	10.86
[**ant** labdent]	1,236	143	65.77
[**distr** cor]	1,236	127	36.69
[**strid** dor]	1,236	116	34.01
[**lat**]	336	44	10.45
[**dor**]	62	13	3.68

TABLE B.36: Samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for all *sufficient* features and selected one for each *insufficient* feature for FE in Chukchi, right-to-left, learned k, with SEAL.

B.4 Polish

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**voice** *son*]	223	201	219	196	201	217	223	217	225	221
[**son**]	39	34	26	28	45	25	9	30	33	3
[**cons**]	39	42	34	44	25	28	33	32	3	4
[**cont**]	24	25	26	7	27	32	31	25	31	35
[**cor**]	28	40	43	32	33	36	40	34	22	39
[**ant**]	165	152	126	160	170	140	168	158	187	145
[**dor**]	28	39	34	26	26	35	31	33	36	33
[**lab**]	27	16	3	36	3	31	27	41	25	44
[**distr**]	39	38	3	23	41	48	37	33	31	29
[**delrel**]	200	159	124	114	117	141	151	133	102	224
[**nasal**]	3	42	42	29	37	43	3	36	27	30
[**lat**]	33	3	46	30	47	31	3	34	36	45
$[^{\textbf{high}\ \ son}_{voice\ \ low}]$	7,310	6,560	8,180	6,060	7,780	7,620	6,620	8,220	5,500	7,340
[**back**]	23	4	37	35	22	42	22	46	27	27
[**low**]	145	169	165	187	157	125	4	152	198	230
$[^{\textbf{tense}\ \ son}_{voice\ \ round}]$	6,870	8,220	7,780	6,100	7,180	8,130	8,100	6,300	8,140	6,900
[**round**]	144	163	150	161	178	4	126	172	166	104
$\lvert\mathbf{M}_F\rvert$	8,185	8,371	8,321	7,669	8,263	8,350	8,336	8,333	8,319	8,200

TABLE B.37: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for each *insufficient* feature, across 10 runs, for FD and OR in Polish, left-to-right, set $k = 3$.

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**voice** son]	231	296	4	262	262	194	299	248	274	257
[**son**]	25	22	26	32	26	32	34	35	5	43
[**cons**]	23	30	27	42	40	46	33	40	50	27
[**cont**]	29	24	30	31	34	29	3	28	24	29
[**cor**]	41	31	37	33	30	31	39	41	49	25
[**ant**]	150	173	171	113	143	150	200	127	144	272
[**dor**]	24	42	32	38	3	37	38	30	33	36
[**lab**]	38	35	42	26	50	40	35	30	44	3
[**distr**]	36	35	3	34	33	27	34	33	25	34
[**delrel**]	168	132	232	211	151	195	126	165	118	137
[**nasal**]	5	38	28	28	25	27	32	32	44	25
[**lat**]	30	3	28	19	50	28	36	30	39	41
$\left[\begin{smallmatrix}\textbf{high} & son\\ voice & low\end{smallmatrix}\right]$	6,860	8,100	7,500	6,380	7,980	6,700	7,060	8,100	7,060	7,860
[**back**]	31	27	29	33	27	24	33	35	29	3
[**low**]	155	218	147	152	154	208	207	111	187	159
$\left[\begin{smallmatrix}\textbf{tense} & son\\ voice & round\end{smallmatrix}\right]$	6,180	6,580	7,340	7,580	7,700	6,380	7,340	7,420	7,620	6,380
[**round**]	221	123	128	145	208	221	135	243	163	147
$\lvert\mathbf{M}_F\rvert$	7,957	8,317	8,293	8,191	8,316	7,985	8,171	8,351	8,223	8,310

TABLE B.38: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for each *insufficient* feature, across 10 runs, for FD and OR in Polish, right-to-left, set $k = 3$.

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**voice** son]	203	159	211	211	209	165	223	223	139	191
[**son**]	2	2	2	2	2	2	2	2	2	2
[**cons**]	2	2	2	2	2	2	2	2	2	2
[**cont**]	2	2	2	2	2	2	2	2	2	2
[**cor**]	2	2	2	2	2	2	2	2	2	2
[**ant**]	3	3	3	3	3	3	3	3	3	3
[**dor**]	2	2	2	2	2	2	2	2	2	2
[**lab**]	2	2	2	2	2	2	2	2	2	2
[**distr**]	2	2	2	2	2	2	2	2	2	2
[**delrel**]	3	3	3	3	3	3	3	3	3	3
[**nasal**]	2	2	2	2	2	2	2	2	2	2
[**lat**]	2	2	2	2	2	2	2	2	2	2
$\left[\begin{smallmatrix}\textbf{high} & son\\ voice & low\end{smallmatrix}\right]$	7,780	8,060	7,780	7,380	4,540	6,740	8,220	5,020	6,780	7,540
[**back**]	2	2	2	2	2	2	2	2	2	2
[**low**]	3	3	3	3	3	3	3	3	3	3
$\left[\begin{smallmatrix}\textbf{tense} & son\\ voice & round\end{smallmatrix}\right]$	7,220	6,980	6,620	7,580	8,220	8,100	7,460	7,460	6,460	5,660
[**round**]	3	3	3	3	3	3	3	3	3	3
$\lvert\mathbf{M}_F\rvert$	8,179	8,222	8,122	8,160	8,234	8,221	8,249	7,880	7,868	8,002

TABLE B.39: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for each *insufficient* feature, across 10 runs, for FD and OR in Polish, left-to-right, learned k.

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**voice** son]	108	156	110	51	84	72	70	108	170	78
[**son**]	2	2	2	2	2	2	2	2	2	2
[**cons**]	2	2	2	2	2	2	2	2	2	2
[**cont**]	2	2	2	2	2	2	2	2	2	2
[**cor**]	2	2	2	2	2	2	2	2	2	2
[**ant**]	3	3	3	3	3	3	3	3	3	3
[**dor**]	2	2	2	2	2	2	2	2	2	2
[**lab**]	2	2	2	2	2	2	2	2	2	2
[**distr**]	2	2	2	2	2	2	2	2	2	2
[**delrel**]	3	3	3	3	3	3	3	3	3	3
[**nasal**]	2	2	2	2	2	2	2	2	2	2
[**lat**]	2	2	2	2	2	2	2	2	2	2
$\left[\begin{smallmatrix}\textbf{high} & son\\ voice & low\end{smallmatrix}\right]$	8,180	8,020	6,500	7,180	7,820	7,180	6,620	6,140	7,340	6,580
[**back**]	2	2	2	2	2	2	2	2	2	2
[**low**]	3	3	3	3	3	3	3	3	3	3
$\left[\begin{smallmatrix}\textbf{tense} & son\\ voice & round\end{smallmatrix}\right]$	7,380	6,900	7,340	7,780	7,620	7,940	6,340	6,220	7,380	7,500
[**round**]	3	3	3	3	3	3	3	3	3	3
$\lvert\mathbf{M}_F\rvert$	8,239	8,210	8,007	8,182	8,203	8,201	7,808	7,625	8,134	8,073

TABLE B.40: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for each *insufficient* feature, across 10 runs, for FD and OR in Polish, right-to-left, learned k.

Feature set Φ	$\lvert S_{(\Phi,\phi)}\rvert$	Avg. $\lvert M_{(\Phi,\phi)}\rvert$	SD
[**voice** son]	336	193	29.18
[**son**]	62	2	0
[**cons**]	62	2	0
[**cont**]	62	2	0
[**cor**]	62	2	0
[**ant**]	336	3	0
[**dor**]	62	2	0
[**lab**]	62	2	0
[**distr**]	62	2	0
[**delrel**]	336	3	0
[**nasal**]	54	2	0
[**lat**]	62	2	0
[**high** voice son low]	8,250	6,984	1,262.72
[**back**]	62	2	0
[**low**]	336	3	0
[**tense** voice son round]	8,250	7,176	776.65
[**round**]	336	3	0

TABLE B.41: Samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for all *sufficient* features and selected one for the *insufficient* features for FD and OR in Polish, left-to-right, learned k.

Feature set Φ	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD
[**voice** son]	336	101	38.13
[**son**]	62	2	0
[**cons**]	62	2	0
[**cont**]	62	2	0
[**cor**]	62	2	0
[**ant**]	336	3	0
[**dor**]	62	2	0
[**lab**]	62	2	0
[**distr**]	62	2	0
[**delrel**]	336	3	0
[**nasal**]	54	2	0
[**lat**]	62	2	0
[**high** voice son low]	8,250	7,156	695.43
[**back**]	62	2	0
[**low**]	336	3	0
[**tense** voice son round]	8,250	7,240	578.20
[**round**]	336	3	0

TABLE B.42: Samples $|S_{(\Phi,\,\phi)}|$, Avg. $|M_{(\Phi,\,\phi)}|$ and standard deviation (SD) for all *sufficient* features and selected one for the *insufficient* features for FD and OR in Polish, right-to-left, learned k.

Feature set Φ	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	**Avg.** $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	**SD**
[**voice** son]	336	214	10.73
[**son**]	62	27	12.77
[**cons**]	62	28	14.37
[**cont**]	62	26	7.70
[**cor**]	62	35	6.34
[**ant**]	336	157	17.27
[**dor**]	62	32	4.33
[**lab**]	62	25	14.26
[**distr**]	62	32	12.28
[**delrel**]	336	147	38.97
[**nasal**]	54	29	14.92
[**lat**]	62	31	15.93
[**high** voice son low]	8,250	7,119	907.70
[**back**]	62	29	12.12
[**low**]	336	153	60.16
[**tense** voice son round]	8,250	7,372	806.83
[**round**]	336	137	51.76

TABLE B.43: Samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for all *sufficient* features and selected one for the *insufficient* features for FD and OR in Polish, left-to-right, set $k = 3$.

Feature set Φ	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	**Avg.** $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	**SD**
[**voice** son]	336	233	85.93
[**son**]	62	28	10.13
[**cons**]	62	36	9.07
[**cont**]	62	26	8.65
[**cor**]	62	36	7.302
[**ant**]	336	164	45.08
[**dor**]	62	31	11.15
[**lab**]	62	34	12.97
[**distr**]	62	29	9.92
[**delrel**]	336	164	38.43
[**nasal**]	54	28	10.19
[**lat**]	62	30	12.89
[**high** voice son low]	8,250	7,360	630.27
[**back**]	62	27	9.10
[**low**]	336	170	33.90
[**tense** voice son round]	8,250	7,052	597.12
[**round**]	336	173	45.04

TABLE B.44: Samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for all *sufficient* features and selected one for the *insufficient* features for FD and OR in Polish, right-to-left, set $k = 3$.

Feature set Φ	$\lvert\Sigma\rvert = 8$		$\lvert\Sigma\rvert = 9$		$\lvert\Sigma\rvert = 10$	
	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$
[**voice** son]	363	241	363	223	363	235
[**son**]	62	32	62	34	62	23
[**cons**]	62	38	62	41	62	31
[**cont**]	62	40	62	27	62	33
[**cor**]	62	7	62	34	336	4
[**ant**]	363	4	363	114	363	135
[**dor**]	62	33	62	3	62	43
[**lab**]	62	47	62	41	336	4
[**distr**]	62	34	62	25	62	29
[**delrel**]	363	141	363	4	363	5
[**nasal**]	62	43	62	29	62	43
[**lat**]	62	32	62	28	62	39
[**high** voice son low]	8,446	7,416	17,892	13,122	17,892	8,122
[**back**]	62	24	62	32	62	37
[**low**]	336	192	336	119	336	182
[**tense** voice son round]	8,446	6,656	17,892	12,482	17,892	17,843
[**round**]	336	162	336	136	336	153
$\lvert\mathbf{M}_F\rvert$	33,352	8,283	59,868	16,272	101,110	17,945

TABLE B.45: Cardinalities $|S_{(\Phi,\,\phi)}|$ and $|M_{(\Phi,\,\phi)}|$ for all *sufficient* features and selected one for each *insufficient* feature, for Σ with 8, 9 and 10 phonemes, for FD and OR in Polish, left-to-right, set $k = 3$.

Feature set Φ	$\lvert\Sigma\rvert = 8$		$\lvert\Sigma\rvert = 9$		$\lvert\Sigma\rvert = 10$	
	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$
[**voice** son]	363	273	363	224	363	277
[**son**]	62	39	62	29	62	43
[**cons**]	62	44	62	3	62	42
[**cont**]	62	30	62	34	62	27
[**cor**]	62	32	62	30	336	152
[**ant**]	363	202	363	134	363	123
[**dor**]	62	36	62	47	62	43
[**lab**]	62	35	62	31	336	201
[**distr**]	62	40	62	38	62	3
[**delrel**]	363	165	363	135	363	102
[**nasal**]	62	29	62	33	62	27
[**lat**]	62	42	62	3	62	41
[**high** voice son low]	8,446	7,896	17,892	14,842	17,892	16,002
[**back**]	62	32	62	41	62	31
[**low**]	336	203	336	101	336	174
[**tense** voice son round]	8,446	6,176	17,892	15,322	17,892	16,522
[**round**]	336	4	336	148	336	150
$\lvert\mathbf{M}_F\rvert$	33,352	8,435	59,868	17,526	101,110	16,846

TABLE B.46: Cardinalities $|S_{(\Phi,\,\phi)}|$ and $|M_{(\Phi,\,\phi)}|$ for all *sufficient* features and selected one for each *insufficient* feature, for Σ with 8, 9 and 10 phonemes, for FD and OR in Polish, right-to-left, set $k = 3$.

Feature set Φ	$\lvert\Sigma\rvert = 8$		$\lvert\Sigma\rvert = 9$		$\lvert\Sigma\rvert = 10$	
	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$
[**voice**, son]	363	127	363	219	363	127
[**son**]	62	2	62	2	62	2
[**cons**]	62	2	62	2	62	2
[**cont**]	62	2	62	2	62	2
[**cor**]	62	2	62	2	336	3
[**ant**]	363	3	363	3	363	3
[**dor**]	62	2	62	2	62	2
[**lab**]	62	2	62	2	336	3
[**distr**]	62	2	62	2	62	2
[**delrel**]	363	3	363	3	363	3
[**nasal**]	62	2	62	2	62	2
[**lat**]	62	2	62	2	62	2
[**high**, voice, son, low]	8,446	8,016	17,892	16,562	17,892	17,122
[**back**]	62	2	62	2	62	2
[**low**]	336	3	336	3	336	3
[**tense**, voice, son, round]	8,446	7,536	17,892	15,722	13,342	17,402
[**round**]	336	3	336	3	336	3
$\lvert\mathbf{M}_F\rvert$	33,352	8,419	59,868	17,705	101,110	17,735

TABLE B.47: Cardinalities $|S_{(\Phi,\,\phi)}|$ and $|M_{(\Phi,\,\phi)}|$ for all *sufficient* features and selected one for each *insufficient* feature, for Σ with 8, 9 and 10 phonemes, for FD and OR in Polish, left-to-right, learned k.

Feature set Φ	$\lvert\Sigma\rvert = 8$		$\lvert\Sigma\rvert = 9$		$\lvert\Sigma\rvert = 10$	
	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$
[**voice** son]	363	98	363	130	363	136
[**son**]	62	2	62	2	62	2
[**cons**]	62	2	62	2	62	2
[**cont**]	62	2	62	2	62	2
[**cor**]	62	2	62	2	336	3
[**ant**]	363	3	363	3	363	3
[**dor**]	62	2	62	2	62	2
[**lab**]	62	2	62	2	336	3
[**distr**]	62	2	62	2	62	2
[**delrel**]	363	3	363	3	363	3
[**nasal**]	62	2	62	2	62	2
[**lat**]	62	2	62	2	62	2
[**high** voice son low]	8,446	7,296	17,892	16,142	17,892	16,842
[**back**]	62	2	62	2	62	2
[**low**]	336	3	336	3	336	3
[**tense** voice son round]	8,446	6,856	17,892	15,322	13,342	16,702
[**round**]	336	3	336	3	336	3
$\lvert\mathbf{M}_F\rvert$	33,352	8,186	59,868	17,367	101,110	17,634

TABLE B.48: Cardinalities $|S_{(\Phi,\,\phi)}|$ and $|M_{(\Phi,\,\phi)}|$ for all *sufficient* features and selected one for each *insufficient* feature, for Σ with 8, 9 and 10 phonemes, for FD and OR in Polish, right-to-left, learned k.

B.4.1 Polish: SEAL

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**voice** *son*]	221	185	209	243	266	113	250	214	252	317
[**son**]	2	2	2	2	2	2	2	2	2	2
[**cons**]	2	2	2	2	2	2	2	2	2	2
[**cont**]	2	2	2	2	2	2	2	2	2	2
[**cor**]	2	2	2	2	2	2	2	2	2	2
[**ant**]	3	3	3	3	3	3	3	3	3	3
[**dor**]	2	2	2	2	2	2	2	2	2	2
[**lab**]	2	2	2	2	2	2	2	2	2	2
[**distr**]	2	2	2	2	2	2	2	2	2	2
[**delrel**]	3	3	3	3	3	3	3	3	3	3
[**nasal**]	2	2	2	2	2	2	2	2	2	2
[**lat**]	2	2	2	2	2	2	2	2	2	2
$\left[\begin{smallmatrix}\textbf{high} & son\\ voice & low\end{smallmatrix}\right]$	6,686	7,586	8,166	7,746	8,106	6,586	8,026	8,106	8,206	7,046
[**back**]	2	2	2	2	2	2	2	2	2	2
[**low**]	3	3	3	3	3	3	3	3	3	3
$\left[\begin{smallmatrix}\textbf{tense} & son\\ voice & round\end{smallmatrix}\right]$	7,426	8,126	7,686	6,046	7,706	8,166	8,246	8,066	7,986	7,706
[**round**]	3	3	3	3	3	3	3	3	3	3
$\lvert\mathbf{M}_F\rvert$	8,089	8,238	8,245	8,106	8,241	8,230	8,250	8,248	8,242	8,171

TABLE B.49: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for each *insufficient* feature, across 10 runs, for FD and OR in Polish, left-to-right, learned k, with SEAL.

Feature set Φ	$\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$									
	1	2	3	4	5	6	7	8	9	10
[**voice** *son*]	3	3	3	3	3	3	3	3	3	3
[**son**]	2	2	2	2	2	2	2	2	2	2
[**cons**]	2	2	2	2	2	2	2	2	2	2
[**cont**]	2	2	2	2	2	2	2	2	2	2
[**cor**]	2	2	2	2	2	2	2	2	2	2
[**ant**]	3	3	3	3	3	3	3	3	3	3
[**dor**]	2	2	2	2	2	2	2	2	2	2
[**lab**]	2	2	2	2	2	2	2	2	2	2
[**distr**]	2	2	2	2	2	2	2	2	2	2
[**delrel**]	3	3	3	3	3	3	3	3	3	3
[**nasal**]	2	2	2	2	2	2	2	2	2	2
[**lat**]	2	2	2	2	2	2	2	2	2	2
$\left[\begin{smallmatrix}\textbf{high} & son\\ voice & low\end{smallmatrix}\right]$	12	22	79	59	26	146	46	21	97	50
[**back**]	2	2	2	2	2	2	2	2	2	2
[**low**]	3	3	3	3	3	3	3	3	3	3
$\left[\begin{smallmatrix}\textbf{tense} & son\\ voice & round\end{smallmatrix}\right]$	29	35	93	26	50	93	70	69	8	68
[**round**]	3	3	3	3	3	3	3	3	3	3
$\lvert\mathbf{M}_F\rvert$	35	51	165	79	70	233	109	84	99	112

TABLE B.50: Cardinalities $|M_{(\Phi,\,\phi)}|$ and $|M_F|$ for all *sufficient* features and selected one for each *insufficient* feature, across 10 runs, for FD and OR in Polish, right-to-left, learned k, with SEAL.

Feature set Φ	$\lvert\mathbf{S}_{(\Phi,\phi)}\rvert$	Avg. $\lvert\mathbf{M}_{(\Phi,\phi)}\rvert$	SD
[**voice** son]	336	227	54.08
[**son**]	62	2	0
[**cons**]	62	2	0
[**cont**]	62	2	0
[**cor**]	62	2	0
[**ant**]	336	3	0
[**dor**]	62	2	0
[**lab**]	62	2	0
[**distr**]	62	2	0
[**delrel**]	336	3	0
[**nasal**]	54	2	0
[**lat**]	62	2	0
[**high** voice son low]	8,250	7,626	629.50
[**back**]	62	2	0
[**low**]	336	3	0
[**tense** voice son round]	8,250	7,716	643.00
[**round**]	336	3	0

TABLE B.51: Samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for all *sufficient* features and selected one for the *insufficient* features for FD and OR in Polish, left-to-right, learned k.

Feature set Φ	$\lvert S_{(\Phi,\phi)}\rvert$	Avg. $\lvert M_{(\Phi,\phi)}\rvert$	SD
[**voice** son]	336	3	0
[**son**]	62	2	0
[**cons**]	62	2	0
[**cont**]	62	2	0
[**cor**]	62	2	0
[**ant**]	336	3	0
[**dor**]	62	2	0
[**lab**]	62	2	0
[**distr**]	62	2	0
[**delrel**]	336	3	0
[**nasal**]	54	2	0
[**lat**]	62	2	0
[**high** voice son low]	8,250	56	41.73
[**back**]	62	2	0
[**low**]	336	3	0
[**tense** voice son round]	8,250	54	29.10
[**round**]	336	3	0

TABLE B.52: Samples $\lvert S_{(\Phi,\,\phi)}\rvert$, Avg. $\lvert M_{(\Phi,\,\phi)}\rvert$ and standard deviation (SD) for all *sufficient* features and selected one for the *insufficient* features for FD and OR in Polish, right-to-left, learned k, with SEAL.

Appendix C

C.1 Koasati

Feature	a	b	f	h	i	k	l	ŋ	n	o	p	s	t	t͡ʃ	w	y	ø
cons	-	+	+	-	-	+	+	+	+	-	+	+	+	+	-	-	-
son	+	-	-	-	+	-	+	+	+	+	-	-	-	-	+	+	+
cont	+	-	+	+	+	-	+	-	-	+	-	+	-	-	+	+	+
delrel	0	-	+	+	0	-	0	0	0	0	-	+	-	+	0	0	0
approx	0	-	-	-	0	-	+	-	-	0	-	-	-	-	+	0	0
tap	0	-	-	-	0	-	-	-	-	0	-	-	-	-	-	0	0
trill	0	-	-	-	0	-	-	-	-	0	-	-	-	-	-	0	0
nasal	-	-	-	-	-	-	-	+	+	-	-	-	-	-	-	-	-
voice	+	+	-	-	+	-	+	+	+	+	-	-	-	-	+	+	+
spreadgl	0	-	-	+	0	-	-	-	-	0	-	-	-	-	-	0	0
constrgl	0	-	-	-	0	-	-	-	-	0	-	-	-	-	-	0	0
lab	0	+	+	-	0	-	-	0	-	0	+	-	-	-	+	0	0
round	-	-	-	-	-	-	-	-	-	+	-	-	-	-	+	+	+
labdent	0	-	+	-	0	-	-	-	-	0	-	-	-	-	-	0	0
cor	0	-	-	-	0	-	+	0	+	0	-	+	+	+	-	0	0
ant	0	0	0	0	0	0	+	0	+	0	0	+	+	-	0	0	0
distr	0	0	0	0	0	0	-	0	-	0	0	-	-	+	0	0	0
strid	0	0	0	0	0	0	-	0	-	0	0	+	-	+	0	0	0
lat	0	-	-	-	0	-	+	-	-	0	-	-	-	-	-	0	0
dor	0	-	-	-	0	+	-	0	-	0	-	-	-	-	+	0	0
high	-	0	0	0	+	+	0	0	0	-	0	0	0	0	+	+	-
low	+	0	0	0	-	-	0	0	0	-	0	0	0	0	-	-	-
front	-	0	0	0	+	0	0	0	0	-	0	0	0	0	-	+	+
back	-	0	0	0	-	0	0	0	0	+	0	0	0	0	+	-	-
tense	0	0	0	0	+	0	0	0	0	+	0	0	0	0	+	+	+

TABLE C.1: Sound inventory in Koasati.

Feature	Sufficient set Φ	k	Direction
cons	[cons]	1	BOTH
son	[son]	1	BOTH
cont	[cont]	1	BOTH
delrel	[delrel]	1	BOTH
approx	[approx]	1	BOTH
tap	[tap]	1	BOTH
trill	[trill]	1	BOTH
nasal	[nasal]	1	BOTH
voice	[voice]	1	BOTH
spreadgl	[spreadgl]	1	BOTH
constrgl	[constrgl]	1	BOTH
lab	[lab, tap]	2	BOTH
round	[round]	1	BOTH
labdent	[labdent]	1	BOTH
cor	[cor, tap]	2	BOTH
ant	[ant, nasal]	2	BOTH
distr	[distr, nasal]	2	BOTH
strid	[strid, nasal]	2	BOTH
lat	[lat]	1	BOTH
dor	[dor, ant, tap]	2	BOTH
high	[high, delrel, nasal]	2	BOTH
low	[low, ant, cor]	2	BOTH
front	[front, delrel, nasal]	2	BOTH
back	[back, ant, son]	2	BOTH
tense	[tense]	1	BOTH

TABLE C.2: Sufficient feature sets, learned memory values, and processing directions for Koasati.

Feature	k	$\lvert Q\rvert$	$\lvert\delta^*\rvert$	$\lvert\Sigma_\Phi\rvert$	Σ_Φ	Segments matched
cons	1	3	4	4	[− cons]	a, h, i, o, w, y, ø
					[+ cons]	b, f, k, l, N, n, p, s, t,t͡S
					[<]	∅
					[>]	∅
son	1	3	4	4	[+ son]	a, i, l, N, n, o, w, y, ø
					[− son]	b, f, h, k, p, s, t, t͡ʃ
					[<]	∅
					[>]	∅
cont	1	3	4	4	[+ cont]	a, f, h, i, l, o, s, w, y, ø
					[− cont]	b, k, N, n, p, t, t͡ʃ
					[<]	∅
					[>]	∅
delrel	1	3	5	5	[0 delrel]	a, i, l, N, n, o, w, y, ø
					[− delrel]	b, k, p, t
					[+ delrel]	f, h, s, t͡ʃ
					[<]	∅
					[>]	∅
approx	1	3	5	5	[0 approx]	a, i, o, y, ø
					[− approx]	b, f, h, k, N, n, p, s, t, t͡ʃ
					[+ approx]	l, w
					[<]	∅
					[>]	∅
tap	1	3	4	4	[0 tap]	a, i, o, y, ø
					[− tap]	b, f, h, k, N, n, p, s, t, t͡ʃ
					[<]	∅
					[>]	∅

TABLE C.3: Koasati: full segment partitions for single-feature sufficient sets (both directions), including automaton size and sigma (as reported in file).

Feature	k	$\lvert Q\rvert$	$\lvert\delta^*\rvert$	$\lvert\Sigma_\Phi\rvert$	Σ_Φ	Segments matched
trill	1	3	5	5	[0 trill]	a, i, o, y, ø
					[− trill]	b, f, h, k, N, n, p, s, t, t͡ʃ
					[+ trill]	r
					[<]	∅
					[>]	∅
nasal	1	3	4	4	[− nasal]	a, b, f, h, i, k, l, o, p, s, t, t͡ʃ, w, y, ø
					[+ nasal]	N, n
					[<]	∅
					[>]	∅
voice	1	3	4	4	[+ voice]	a, b, i, l, N, n, o, w, y, ø
					[− voice]	f, h, k, p, s, t, t͡ʃ
					[<]	∅
					[>]	∅
spreadgl	1	3	5	5	[0 spreadgl]	a, i, o, y, ø
					[− spreadgl]	b, f, k, l, N, n, p, s, t, t͡ʃ, w
					[+ spreadgl]	h
					[<]	∅
					[>]	∅
constrgl	1	3	4	4	[0 constrgl]	a, i, o, y, ø
					[− constrgl]	b, f, h, k, l, N, n, p, s, t, t͡ʃ, w
					[<]	∅
					[>]	∅
round	1	3	4	4	[− round]	a, b, f, h, i, k, l, N, n, p, s, t, t͡ʃ
					[+ round]	o, w, y, ø
					[<]	∅
					[>]	∅

TABLE C.4: Koasati: full segment partitions for single-feature sufficient sets (both directions), including automaton size and sigma (as reported in file).

Feature	k	$\lvert Q\rvert$	$\lvert\delta^*\rvert$	$\lvert\Sigma_\Phi\rvert$	Σ_Φ	Segments matched
labdent	1	3	5	5	[0 labdent]	a, i, o, y, ø
					[− labdent]	b, h, k, l, N, n, p, s, t, t͡ʃ, w
					[+ labdent]	f
					[<]	∅
					[>]	∅
lat	1	3	5	5	[0 lat]	a, i, o, y, ø
					[− lat]	b, f, h, k, N, n, p, s, t, t͡ʃ, w
					[+ lat]	l
					[<]	∅
					[>]	∅
tense	1	3	4	4	[0 tense]	a, b, f, h, k, l, N, n, p, r, s, t, t͡ʃ
					[+ tense]	i, o, w, y, ø
					[<]	∅
					[>]	∅

TABLE C.5: Koasati: full segment partitions for single-feature sufficient sets (both directions), including automaton size and sigma (as reported in file).

Feature	k	$\lvert Q\rvert$	$\lvert\delta^*\rvert$	$\lvert\Sigma_\Phi\rvert$	Σ_Φ	Segments matched
lab	2	7	26	6	$\begin{bmatrix} 0 & \text{lab} \\ 0 & \text{tap} \end{bmatrix}$	a, i, o, y, ø
					$\begin{bmatrix} + & \text{lab} \\ - & \text{tap} \end{bmatrix}$	b, f, p, w
					$\begin{bmatrix} - & \text{lab} \\ - & \text{tap} \end{bmatrix}$	h, k, l, n, s, t, $\widehat{\text{t}\int}$
					$\begin{bmatrix} 0 & \text{lab} \\ - & \text{tap} \end{bmatrix}$	N
					$[<]$	∅
					$[>]$	∅
cor	2	7	26	6	$\begin{bmatrix} 0 & \text{cor} \\ 0 & \text{tap} \end{bmatrix}$	a, i, o, y, ø
					$\begin{bmatrix} - & \text{cor} \\ - & \text{tap} \end{bmatrix}$	b, f, h, k, p, w
					$\begin{bmatrix} + & \text{cor} \\ - & \text{tap} \end{bmatrix}$	l, n, s, t, $\widehat{\text{t}\int}$
					$\begin{bmatrix} 0 & \text{cor} \\ - & \text{tap} \end{bmatrix}$	N
					$[<]$	∅
					$[>]$	∅
ant	2	8	37	7	$\begin{bmatrix} 0 & \text{ant} \\ - & \text{nasal} \end{bmatrix}$	a, b, f, h, i, k, o, p, w, y, ø
					$\begin{bmatrix} + & \text{ant} \\ - & \text{nasal} \end{bmatrix}$	l, s, t
					$\begin{bmatrix} 0 & \text{ant} \\ + & \text{nasal} \end{bmatrix}$	N
					$\begin{bmatrix} + & \text{ant} \\ + & \text{nasal} \end{bmatrix}$	n
					$\begin{bmatrix} - & \text{ant} \\ - & \text{nasal} \end{bmatrix}$	$\widehat{\text{t}\int}$
					$[<]$	∅
					$[>]$	∅
distr	2	8	37	7	$\begin{bmatrix} 0 & \text{distr} \\ - & \text{nasal} \end{bmatrix}$	a, b, f, h, i, k, o, p, w, y, ø
					$\begin{bmatrix} - & \text{distr} \\ - & \text{nasal} \end{bmatrix}$	l, s, t
					$\begin{bmatrix} 0 & \text{distr} \\ + & \text{nasal} \end{bmatrix}$	N
					$\begin{bmatrix} - & \text{distr} \\ + & \text{nasal} \end{bmatrix}$	n
					$\begin{bmatrix} + & \text{distr} \\ - & \text{nasal} \end{bmatrix}$	$\widehat{\text{t}\int}$
					$[<]$	∅
					$[>]$	∅

TABLE C.6: Koasati: full segment partitions for multi-feature sufficient sets (both directions), with each sigma element shown as a value–feature matrix.

Feature	k	$\lvert Q\rvert$	$\lvert\delta^*\rvert$	$\lvert\Sigma_\Phi\rvert$	Σ_Φ	Segments matched
strid	2	8	37	7	$\begin{bmatrix} 0 & \text{strid} \\ - & \text{nasal} \end{bmatrix}$	a, b, f, h, i, k, o, p, w, y, ø
					$\begin{bmatrix} - & \text{strid} \\ - & \text{nasal} \end{bmatrix}$	l, t
					$\begin{bmatrix} 0 & \text{strid} \\ + & \text{nasal} \end{bmatrix}$	N
					$\begin{bmatrix} - & \text{strid} \\ + & \text{nasal} \end{bmatrix}$	n
					$\begin{bmatrix} + & \text{strid} \\ - & \text{nasal} \end{bmatrix}$	s, $\widehat{\text{t}\int}$
					$\begin{bmatrix} < \end{bmatrix}$	$\emptyset$
					$\begin{bmatrix} > \end{bmatrix}$	$\emptyset$
dor	2	9	50	8	$\begin{bmatrix} 0 & \text{dor} \\ 0 & \text{ant} \\ 0 & \text{tap} \end{bmatrix}$	a, i, o, y, ø
					$\begin{bmatrix} - & \text{dor} \\ 0 & \text{ant} \\ - & \text{tap} \end{bmatrix}$	b,f, h, p
					$\begin{bmatrix} + & \text{dor} \\ 0 & \text{ant} \\ - & \text{tap} \end{bmatrix}$	k, w
					$\begin{bmatrix} - & \text{dor} \\ + & \text{ant} \\ - & \text{tap} \end{bmatrix}$	l, n, s, t
					$\begin{bmatrix} 0 & \text{dor} \\ 0 & \text{ant} \\ - & \text{tap} \end{bmatrix}$	N
					$\begin{bmatrix} - & \text{dor} \\ - & \text{ant} \\ - & \text{tap} \end{bmatrix}$	$\widehat{\text{t}\int}$
					$\begin{bmatrix} < \end{bmatrix}$	$\emptyset$
					$\begin{bmatrix} > \end{bmatrix}$	$\emptyset$
high	2	10	65	9	$\begin{bmatrix} - & \text{high} \\ 0 & \text{delrel} \\ - & \text{nasal} \end{bmatrix}$	a, o, ø
					$\begin{bmatrix} 0 & \text{high} \\ - & \text{delrel} \\ - & \text{nasal} \end{bmatrix}$	b, p, t
					$\begin{bmatrix} 0 & \text{high} \\ + & \text{delrel} \\ - & \text{nasal} \end{bmatrix}$	f, h, s, $\widehat{\text{t}\int}$
					$\begin{bmatrix} + & \text{high} \\ 0 & \text{delrel} \\ - & \text{nasal} \end{bmatrix}$	i, w, y
					$\begin{bmatrix} + & \text{high} \\ - & \text{delrel} \\ - & \text{nasal} \end{bmatrix}$	k
					$\begin{bmatrix} 0 & \text{high} \\ 0 & \text{delrel} \\ - & \text{nasal} \end{bmatrix}$	l
					$\begin{bmatrix} 0 & \text{high} \\ 0 & \text{delrel} \\ + & \text{nasal} \end{bmatrix}$	N, n
					$\begin{bmatrix} < \end{bmatrix}$	$\emptyset$
					$\begin{bmatrix} > \end{bmatrix}$	$\emptyset$

TABLE C.7: Koasati: full segment partitions for multi-feature sufficient sets (both directions), with each sigma element shown as a value–feature matrix.

Feature	k	$\lvert Q\rvert$	$\lvert\delta^*\rvert$	$\lvert\Sigma_\Phi\rvert$	Σ_Φ	Segments matched
low	2	10	65	9	$\begin{bmatrix} + & \text{low} \\ 0 & \text{ant} \\ 0 & \text{cor} \end{bmatrix}$	a
					$\begin{bmatrix} 0 & \text{low} \\ 0 & \text{ant} \\ - & \text{cor} \end{bmatrix}$	b, f, h, p
					$\begin{bmatrix} - & \text{low} \\ 0 & \text{ant} \\ 0 & \text{cor} \end{bmatrix}$	i, o, y, ø
					$\begin{bmatrix} - & \text{low} \\ 0 & \text{ant} \\ - & \text{cor} \end{bmatrix}$	k, w
					$\begin{bmatrix} 0 & \text{low} \\ + & \text{ant} \\ + & \text{cor} \end{bmatrix}$	l, n, s, t
					$\begin{bmatrix} 0 & \text{low} \\ 0 & \text{ant} \\ 0 & \text{cor} \end{bmatrix}$	N
					$\begin{bmatrix} 0 & \text{low} \\ - & \text{ant} \\ + & \text{cor} \end{bmatrix}$	$\widehat{\text{t}\int}$
					$[<]$	∅
					$[>]$	∅
front	2	9	50	8	$\begin{bmatrix} - & \text{front} \\ 0 & \text{delrel} \\ - & \text{nasal} \end{bmatrix}$	a, o, w
					$\begin{bmatrix} 0 & \text{front} \\ - & \text{delrel} \\ - & \text{nasal} \end{bmatrix}$	b, k, p, t
					$\begin{bmatrix} 0 & \text{front} \\ + & \text{delrel} \\ - & \text{nasal} \end{bmatrix}$	f, h, s, $\widehat{\text{t}\int}$
					$\begin{bmatrix} + & \text{front} \\ 0 & \text{delrel} \\ - & \text{nasal} \end{bmatrix}$	i, y, ø
					$\begin{bmatrix} 0 & \text{front} \\ 0 & \text{delrel} \\ - & \text{nasal} \end{bmatrix}$	l
					$\begin{bmatrix} 0 & \text{front} \\ 0 & \text{delrel} \\ + & \text{nasal} \end{bmatrix}$	N, n
					$[<]$	∅
					$[>]$	∅
back	2	10	65	9	$\begin{bmatrix} - & \text{back} \\ 0 & \text{ant} \\ + & \text{son} \end{bmatrix}$	a, i, y, ø
					$\begin{bmatrix} 0 & \text{back} \\ 0 & \text{ant} \\ - & \text{son} \end{bmatrix}$	b, f, h, k, p
					$\begin{bmatrix} 0 & \text{back} \\ + & \text{ant} \\ + & \text{son} \end{bmatrix}$	l, n
					$\begin{bmatrix} 0 & \text{back} \\ 0 & \text{ant} \\ + & \text{son} \end{bmatrix}$	N
					$\begin{bmatrix} + & \text{back} \\ 0 & \text{ant} \\ + & \text{son} \end{bmatrix}$	o, w
					$\begin{bmatrix} 0 & \text{back} \\ + & \text{ant} \\ - & \text{son} \end{bmatrix}$	s, t
					$\begin{bmatrix} 0 & \text{back} \\ - & \text{ant} \\ - & \text{son} \end{bmatrix}$	$\widehat{\text{t}\int}$
					$[<]$	∅
					$[>]$	∅

TABLE C.8: Koasati: full segment partitions for multi-feature sufficient sets (both directions), with each sigma element shown as a value–feature matrix.

C.2 Kerewe

feature	a	b	t͡ʃ	d	e	g	h	i	k	m	n	p	u	ŋ
cons	−	+	+	+	−	+	−	−	+	+	+	+	−	+
son	+	−	−	−	+	−	−	+	−	+	+	−	+	+
cont	+	−	−	−	+	−	+	+	−	−	−	−	+	−
delrel	0	−	+	−	0	−	+	0	−	0	0	−	0	0
approx	0	−	−	−	0	−	−	0	−	−	−	−	0	−
tap	0	−	−	−	0	−	−	0	−	−	−	−	0	−
trill	0	−	−	−	0	−	−	0	−	−	−	−	0	−
nasal	−	−	−	−	−	−	−	−	−	+	+	−	−	+
voice	+	+	−	+	+	+	−	+	−	+	+	−	+	+
spreadgl	0	−	−	−	0	−	+	0	−	−	−	−	0	−
constrgl	0	−	−	−	0	−	−	0	−	−	−	−	0	−
lab	0	+	−	−	0	−	−	0	−	+	−	+	0	−
round	−	−	−	−	−	−	−	−	−	−	−	−	+	−
labdent	0	−	−	−	0	−	−	0	−	−	−	−	0	−
cor	0	−	+	+	0	−	−	0	−	−	+	−	0	−
ant	0	0	−	+	0	0	0	0	0	0	+	0	0	0
distr	0	0	+	−	0	0	0	0	0	0	−	0	0	0
strid	0	0	+	−	0	0	0	0	0	0	−	0	0	0
lat	0	−	−	−	0	−	−	0	−	−	−	−	0	−
dor	0	−	−	−	0	+	−	0	+	−	−	−	0	+
high	−	0	0	0	−	+	0	+	+	0	0	0	+	+
front	−	0	0	0	+	−	0	+	−	0	0	0	−	−
back	−	0	0	0	−	0	0	−	0	0	0	0	+	0
tense	0	0	0	0	+	0	0	+	0	0	0	0	+	0
low	+	0	0	0	−	0	0	−	0	0	0	0	−	0

TABLE C.9: Sound inventory in Kerewe

Feature	Sufficient set Φ	k	Direction
cons	[cons, approx]	2	BOTH
son	[son]	1	BOTH
cont	[cont, approx]	2	BOTH
delrel	[delrel, approx]	2	BOTH
approx	[approx]	1	BOTH
tap	[tap]	1	BOTH
trill	[trill]	1	BOTH
nasal	[nasal]	1	BOTH
voice	[voice]	1	BOTH
spreadgl	[spreadgl]	2	BOTH
constrgl	[constrgl]	1	BOTH
lab	[lab, cons]	2	BOTH
round	[round]	1	BOTH
labdent	[labdent]	1	BOTH
cor	[cor]	1	BOTH
ant	[ant]	1	BOTH
distr	[distr]	1	BOTH
strid	[strid]	1	BOTH
lat	[lat]	1	BOTH
dor	[dor]	1	BOTH
high	[high]	1	BOTH
low	[low]	1	BOTH
front	[front]	1	BOTH
back	[back]	1	BOTH
tense	[tense]	1	BOTH

TABLE C.10: Sufficient feature sets, learned memory values, and processing directions for Kerewe.

C.3 Lumasaaba

Feature	b	β	m	f	t	d	n	l	k	g	ŋ	w	ɲ	N	i	u	e	o	a	c	ɟ
cons	+	+	+	+	+	+	+	+	+	+	+	-	+	+	-	-	-	-	-	+	+
son	-	-	+	-	-	-	+	+	-	-	+	+	+	+	+	+	+	+	+	-	-
cont	-	+	-	+	-	-	-	+	-	-	-	+	-	-	+	+	+	+	+	-	-
delrel	-	+	0	+	-	-	0	0	-	-	0	0	0	0	0	0	0	0	0	-	-
approx	-	-	-	-	-	-	-	+	-	-	-	+	-	-	0	0	0	0	0	-	-
tap	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	-	-
trill	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	-	-
nasal	-	-	+	-	-	-	+	-	-	-	+	-	+	+	-	-	-	-	-	-	-
voice	+	+	+	-	-	+	+	+	-	+	+	+	+	+	+	+	+	+	+	-	+
spreadgl	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	-	-
constrgl	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	-	-
lab	+	+	+	+	-	-	-	-	-	-	-	+	-	0	0	0	0	0	0	-	-
round	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	+	-	+	-	-	-
labdent	-	-	-	+	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	-	-
cor	-	-	-	-	+	+	+	+	-	-	-	-	+	0	0	0	0	0	0	+	+
ant	0	0	0	0	+	+	+	+	0	0	0	0	-	0	0	0	0	0	0	-	-
distr	0	0	0	0	-	-	-	-	0	0	0	0	+	0	0	0	0	0	0	+	+
strid	0	0	0	0	-	-	-	-	0	0	0	0	-	0	0	0	0	0	0	-	-
lat	-	-	-	-	-	-	-	+	-	-	-	-	-	-	0	0	0	0	0	-	-
dor	-	-	-	-	-	-	-	-	+	+	+	+	+	0	0	0	0	0	0	+	+
high	0	0	0	0	0	0	0	0	+	+	+	+	+	0	+	+	-	-	-	+	+
low	0	0	0	0	0	0	0	0	-	-	-	-	-	0	-	-	-	-	+	-	-
front	0	0	0	0	0	0	0	0	0	0	0	-	+	0	+	-	+	-	-	+	+
back	0	0	0	0	0	0	0	0	0	0	0	+	-	0	-	+	-	+	-	-	-
tense	0	0	0	0	0	0	0	0	0	0	0	+	0	0	+	+	+	+	0	0	0

TABLE C.11: Sound inventory in Lumasaaba.

Feature	Sufficient set Φ	k	Direction
cons	[cons]	1	BOTH
son	[son]	1	BOTH
cont	[cont, cons]	2	BOTH
delrel	[delrel, cons]	2	BOTH
approx	[approx]	1	BOTH
tap	[tap]	1	BOTH
trill	[trill]	1	BOTH
nasal	[nasal]	1	BOTH
voice	[voice, cons]	2	BOTH
spreadgl	[spreadgl]	1	BOTH
constrgl	[constrgl]	1	BOTH
lab	[lab, tap]	2	BOTH
round	[round]	1	BOTH
labdent	[labdent]	1	BOTH
cor	[cor, lab, round, tap, tense]	3	BOTH
ant	[ant, low, round]	3	BOTH
distr	[distr, low, round]	3	BOTH
strid	[strid, front]	2	BOTH
lat	[lat]	1	BOTH
dor	[dor, tap]	2	BOTH
high	[high, cons]	2	BOTH
low	[low, cont]	2	BOTH
front	[front, low]	3	BOTH
back	[back, low]	3	BOTH
tense	[tense]	1	BOTH

TABLE C.12: Sufficient feature sets, learned memory values, and processing directions for Lumasaaba.

C.4 Samoan

Feature	p	m	f	v	t	s	n	l	ŋ	ʔ	i	u	e	o	a
cons	+	+	+	+	+	+	+	+	+	+	-	-	-	-	-
son	-	+	-	-	-	-	+	+	+	-	+	+	+	+	+
cont	-	-	+	+	-	+	-	+	-	-	+	+	+	+	+
delrel	-	0	+	+	-	+	0	0	0	-	0	0	0	0	0
approx	-	-	-	-	-	-	-	+	-	-	0	0	0	0	0
tap	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0
trill	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0
nasal	-	+	-	-	-	-	+	-	+	-	-	-	-	-	-
voice	-	+	-	+	-	-	+	+	+	-	+	+	+	+	+
spreadgl	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0
constrgl	-	-	-	-	-	-	-	-	-	+	0	0	0	0	0
lab	+	+	+	+	-	-	-	-	-	-	0	0	0	0	0
round	-	-	-	-	-	-	-	-	-	-	-	+	-	+	-
labdent	-	-	+	+	-	-	-	-	-	-	0	0	0	0	0
cor	-	-	-	-	+	+	+	+	-	-	0	0	0	0	0
ant	0	0	0	0	+	+	+	+	0	0	0	0	0	0	0
distr	0	0	0	0	-	-	-	-	0	0	0	0	0	0	0
strid	0	0	0	0	-	+	-	-	0	0	0	0	0	0	0
lat	-	-	-	-	-	-	-	+	-	-	0	0	0	0	0
dor	-	-	-	-	-	-	-	-	+	-	0	0	0	0	0
high	0	0	0	0	0	0	0	0	+	0	+	+	-	-	-
low	0	0	0	0	0	0	0	0	-	0	-	-	-	-	+
front	0	0	0	0	0	0	0	0	0	0	+	-	+	-	-
back	0	0	0	0	0	0	0	0	0	0	-	+	-	+	-
tense	0	0	0	0	0	0	0	0	0	0	+	+	+	+	0

TABLE C.13: Sound inventory in Samoan.

Feature	Sufficient set Φ	k	Direction
cons	[cons, front]	2	BOTH
son	[son, front]	2	BOTH
cont	[cont, front]	2	BOTH
delrel	[delrel, front]	2	BOTH
approx	[approx, front]	2	BOTH
tap	[tap, front]	2	BOTH
trill	[trill, front]	2	BOTH
nasal	[nasal, front]	2	BOTH
voice	[voice, front]	2	BOTH
spreadgl	[spreadgl, front]	2	BOTH
constrgl	[constrgl, front]	2	BOTH
lab	[lab, front]	2	BOTH
round	[round, front]	2	BOTH
labdent	[labdent, front]	2	BOTH
cor	[cor, front]	2	BOTH
ant	[ant, front]	2	BOTH
distr	[distr, front]	2	BOTH
strid	[strid, front]	2	BOTH
lat	[lat, front]	2	BOTH
dor	[dor, front]	2	BOTH
high	[high, front]	2	BOTH
low	[low, front]	2	BOTH
front	[front]	2	BOTH
back	[back, front]	2	BOTH
tense	[tense, front]	2	BOTH

TABLE C.14: Sufficient feature sets, learned memory values, and processing directions for Samoan.

C.5 Serbo-Croatian

Feature	a	b	d	e	g	i	k	l	m	n	o	p	r	s	t	t͡ʃ	u	v	j	z	ʂ	ʐ
cons	-	+	+	-	+	-	+	+	+	+	-	+	+	+	+	+	-	+	-	+	+	+
son	+	-	-	+	-	+	-	+	+	+	+	-	+	-	-	-	+	-	+	-	-	-
cont	+	-	-	+	-	+	-	+	-	-	+	-	+	+	-	-	+	+	+	+	+	+
delrel	0	-	-	0	-	0	-	0	0	0	0	-	0	+	-	+	0	+	0	+	+	+
approx	0	-	-	0	-	0	-	+	-	-	0	-	+	-	-	-	0	-	+	-	-	-
tap	0	-	-	0	-	0	-	-	-	-	0	-	-	-	-	-	0	-	-	-	-	-
trill	0	-	-	0	-	0	-	-	-	-	0	-	+	-	-	-	0	-	-	-	-	-
nasal	-	-	-	-	-	-	-	-	+	+	-	-	-	-	-	-	-	-	-	-	-	-
voice	+	+	+	+	+	+	-	+	+	+	+	-	+	-	-	-	+	+	+	+	-	+
spreadgl	0	-	-	0	-	0	-	-	-	-	0	-	-	-	-	-	0	-	-	-	-	-
constrgl	0	-	-	0	-	0	-	-	-	-	0	-	-	-	-	-	0	-	-	-	-	-
lab	0	+	-	0	-	0	-	-	+	-	0	+	-	-	-	-	0	+	-	-	-	-
round	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-	-	+	-	-	-	-	-
labdent	0	-	-	0	-	0	-	-	-	-	0	-	-	-	-	-	0	+	-	-	-	-
cor	0	-	+	0	-	0	-	+	-	+	0	-	+	+	+	+	0	-	-	+	+	+
ant	0	0	+	0	0	0	0	+	0	+	0	0	+	+	+	-	0	0	0	+	-	-
distr	0	0	-	0	0	0	0	-	0	-	0	0	-	-	-	+	0	0	0	-	-	-
strid	0	0	-	0	0	0	0	-	0	-	0	0	-	+	-	+	0	0	0	+	+	+
lat	0	-	-	0	-	0	-	+	-	-	0	-	-	-	-	-	0	-	-	-	-	-
dor	0	-	-	0	+	0	+	-	-	-	0	-	-	-	-	-	0	-	+	-	-	-
high	-	0	0	-	+	+	+	0	0	0	-	0	0	0	0	0	+	0	+	0	0	0
low	+	0	0	-	-	-	-	0	0	0	-	0	0	0	0	0	-	0	-	0	0	0
front	-	0	0	+	0	+	0	0	0	0	-	0	0	0	0	0	-	0	+	0	0	0
back	-	0	0	-	0	-	0	0	0	0	+	0	0	0	0	0	+	0	-	0	0	0
tense	0	0	0	+	0	+	0	0	0	0	+	0	0	0	0	0	+	0	+	0	0	0

TABLE C.15: Sound inventory in Serbo-Croatian.

Feature	Sufficient set Φ	k	Direction
cons	[cons, lat]	3	BOTH
son	[son, cons]	3	BOTH
cont	[cont, cons]	3	BOTH
delrel	[delrel, cons]	3	BOTH
approx	[approx, trill]	3	BOTH
tap	[tap, lat]	3	BOTH
trill	[trill, approx]	3	BOTH
nasal	[nasal, cons]	3	BOTH
voice	[voice, cons]	3	BOTH
spreadgl	[spreadgl, lat]	3	BOTH
constrgl	[constrgl, lat]	3	BOTH
lab	[lab, lat]	3	BOTH
round	[round, lat]	3	BOTH
labdent	[labdent, lat]	3	BOTH
cor	[cor, lat]	3	BOTH
ant	[ant, lat]	3	BOTH
distr	[distr, lat]	3	BOTH
strid	[strid, lat]	3	BOTH
lat	[lat]	3	BOTH
dor	[dor, lat]	3	BOTH
high	[high, lat]	3	L2R
low	[low, lat]	3	BOTH
front	[front, lat]	3	BOTH
back	[back, lat]	3	BOTH
tense	[tense, lat]	3	BOTH

TABLE C.16: Sufficient feature sets, learned memory values, and processing directions for Serbo-Croatian.

Learner-selected	Manually imposed
[**cons**, lat]	[**cons**, lat]
[**son**, cons]	[**son**, lat]
[**cont**, cons]	[**cont**, lat]
[**delrel**, cons]	[**delrel**, lat]
[**approx**, trill]	[**approx**, lat]
[**tap**, lat]	[**tap**, lat]
[**trill**, approx]	[**trill**, lat]
[**nasal**, cons]	[**nasal**, lat]
[**voice**, cons]	[**voice**, lat]
[**spreadgl**, lat]	[**spreadgl**, lat]
[**constrgl**, lat]	[**constrgl**, lat]
[**lab**, lat]	[**lab**, lat]
[**round**, lat]	[**round**, lat]
[**labdent**, lat]	[**labdent**, lat]
[**cor**, lat]	[**cor**, lat]
[**ant**, lat]	[**ant**, lat]
[**distr**, lat]	[**distr**, lat]
[**strid**, lat]	[**strid**, lat]
[**lat**]	[**lat**]
[**dor**, lat]	[**dor**, lat]
[**high**, lat]	[**high**, lat]
[**low**, lat]	[**low**, lat]
[**front**, lat]	[**front**, lat]
[**back**, lat]	[**back**, lat]
[**tense**, lat]	[**tense**, lat]

TABLE C.17: Feature specifications used in the two Serbo-Croatian cross-validation conditions. The base feature is shown in bold.

#	UR	SR
1	d͡z ɛ i v	t͡s ɛ i v
2	b ɔ r	p u r
3	d͡ʑ i	t͡ɕ i
4	d͡z ɛ i v ɔ p	t͡s ɛ i v ɔ p
5	d͡ʑ ɔ x	t͡ɕ u x
6	v ɔ m	f u m
7	d͡ʑ j u p	t͡ɕ j u p
8	v ɔ i s u t a t	f u i s u t a t
9	v ɔ n a p	f u n a p
10	d͡z ɛ j	t͡s ɛ j
11	z a r	s a r
12	v a b	f a b
13	ʐ a k ɔ p	ʂ a k ɔ p
14	v ɔ m ɔ d	f u m ɔ d
15	d͡ʑ ɛ j	t͡ɕ ɛ j
16	d͡ʑ j ɨ ʐ p	t͡ɕ j ɨ ʐ p
17	ʑɛ v	ɕɛ v
18	d͡z ɛ j z	t͡s ɛ j z
19	d͡z ɛ i v ɔ p ɔ	t͡s ɛ i v ɔ p ɔ
20	j ɔ v s	j u v s
21	v ɔ i s i m	f u i s i m
22	d͡ʑ ɛ j ɔ p	t͡ɕ ɛ j ɔ p
23	ʐ t a p	ʂ t a p
24	v ɔ k ɛ i v ɔ w t͡ʂ	f u k ɛ i v ɔ w t͡ʂ
25	v ɔ k d a i d͡z	f u k d a i d͡z
26	l ɔ b	l u b
27	ʑu b	ɕu b
28	ʐ ɔ w ɔ p	ʂ u w ɔ p
29	v a t͡s ɔ z	f a t͡s ɔ z
30	ʐ t a p ɔ p	ʂ t a p ɔ p
31	ʐ ɛ l	ʂ ɛ l
32	v ɔ k s ɛ i p	f u k s ɛ i p
33	r ɔ g	r u g
34	j ɔ k ɔ p	j u k ɔ p
35	g ɔ̃ i t͡s ɔ p	k ɔ̃ i t͡s ɔ p
36	v ɔ g ɔ̃ i t͡s ɔ p	f u g ɔ̃ i t͡s ɔ p
37	v ɛ ʐ d	f ɛ ʐ d
38	ʐ ɔ w v	ʂ u w v
39	v ɔ k m ɔ d	f u k m ɔ d
40	d͡ʑ j a n z	t͡ɕ j a n z
41	ʑɛ l a n z	ɕɛ l a n z
42	d ɔ x ɔ m a s	t u x ɔ m a s
43	v ɔ d ɔ x ɔ m a s	f u d ɔ x ɔ m a s
44	ʐ ɛ i b	ʂ ɛ i b
45	j ɔ t͡s	j u t͡s
46	v ɔ d a i b ɔ	f u d a i b ɔ
47	v ɔ k t͡s ɔ l k	f u k t͡s ɔ l k
48	d͡ʑ ɛ j ɨ ʐ p	t͡ɕ ɛ j ɨ ʐ p
49	v ɔ k z a r b ɔ	f u k z a r b ɔ
50	d͡ʑ j ɨ v	t͡ɕ j ɨ v
51	w ɔ t͡s	w u t͡s

TABLE C.18: Non-identical training pairs for Polish in fold 5.

Fold	Type	UR	Gold SR	Failed ϕ
Feature combinations selected by the learner: L2R				
1/5	non-id.	dobr	dobar	ant, cor, distr, lab, strid
1/5	non-id.	obl	obao	lab
1/5	id.	mukla	mukla	low
1/5	id.	uska	uska	high
2/5	non-id.	mukl	mukao	dor, high, low
2/5	non-id.	ustal	ustao	front, low, tense
2/5	id.	zelena	zelena	low
2/5	id.	zeleni	zeleni	low
2/5	id.	vaz̩ni	vaz̩ni	low
3/5	non-id.	blizk	blizak	ant, cor, delrel, distr, dor, high, low, son, strid, voice
3/5	id.	visoki	visoki	nasal
3/5	id.	duboko	duboko	dor, low, son
3/5	id.	kratko	kratko	ant, cor, delrel, distr, high, strid, voice
3/5	id.	duboka	duboka	son
3/5	id.	bogat	bogat	dor
3/5	id.	uski	uski	ant, cor, delrel, distr, high, strid, voice
3/5	id.	beli	beli	low

Fold	Type	UR	Gold SR	Failed ϕ
3/5	id.	mukli	mukli	high
3/5	id.	bliska	bliska	dor, voice
3/5	id.	bodro	bodro	ant, cor, delrel, distr, low, strid
3/5	id.	sitni	sitni	high
4/5	id.	milo	milo	high
4/5	id.	grub	grub	ant, cor, distr, strid
4/5	id.	vaz̩no	vaz̩no	tense
4/5	id.	bogati	bogati	low
5/5	non-id.	mil	mio	high
5/5	non-id.	vaz̩n	vaz̩an	ant
5/5	id.	rapav	rapav	delrel, labdent
5/5	id.	blag	blag	low
5/5	id.	blaga	blaga	low
Feature combinations selected by the learner: R2L				
1/5	non-id.	dobr	dobar	ant, cor, distr, lab, strid
1/5	non-id.	obl	obao	lab
1/5	id.	mukla	mukla	low
1/5	id.	uska	uska	high
2/5	non-id.	mukl	mukao	dor, high, low
2/5	non-id.	ustal	ustao	front, low
2/5	id.	zelena	zelena	low
2/5	id.	zeleni	zeleni	low

Fold	Type	UR	Gold SR	Failed ϕ
2/5	id.	vaz̩ni	vaz̩ni	low
3/5	non-id.	blizk	blizak	ant, cor, delrel, distr, dor, high, low, son, strid, voice
3/5	id.	visoki	visoki	nasal
3/5	id.	duboko	duboko	dor, low
3/5	id.	kratko	kratko	ant, cor, delrel, distr, high, strid, voice
3/5	id.	duboka	duboka	son
3/5	id.	bogat	bogat	dor
3/5	id.	uski	uski	ant, cor, delrel, distr, high, strid, voice
3/5	id.	beli	beli	low
3/5	id.	mukli	mukli	high
3/5	id.	bliska	bliska	dor, voice
3/5	id.	bodro	bodro	ant, cor, delrel, distr, low, strid
3/5	id.	sitni	sitni	high
4/5	id.	milo	milo	high
4/5	id.	grub	grub	ant, cor, distr, strid
5/5	non-id.	mil	mio	high
5/5	non-id.	vaz̩n	vaz̩an	ant

Fold	Type	UR	Gold SR	Failed ϕ
5/5	id.	rapav	rapav	delrel, labdent
5/5	id.	obla	obla	low
5/5	id.	podla	podla	low
Manually imposed phonological feature combinations: L2R				
1/5	non-id.	dobr	dobar	ant, cor, distr, lab, strid
1/5	non-id.	obl	obao	lab
1/5	id.	mukla	mukla	low
1/5	id.	uska	uska	high
2/5	non-id.	mukl	mukao	dor, high, low, voice
2/5	non-id.	ustal	ustao	front, low, tense
2/5	non-id.	bel	beo	nasal
2/5	id.	zelena	zelena	low, nasal
2/5	id.	zeleni	zeleni	low
2/5	id.	vaz̨ni	vaz̨ni	low
3/5	non-id.	blizk	blizak	ant, cor, delrel, distr, dor, high, low, son, strid, voice
3/5	non-id.	uzk	uzak	voice
3/5	id.	duboko	duboko	dor, low, nasal, son
3/5	id.	kratko	kratko	ant, cor, delrel, distr, high, strid, voice

Fold	Type	UR	Gold SR	Failed ϕ
3/5	id.	bogat	bogat	dor, son
3/5	id.	uski	uski	ant, cor, delrel, distr, high, strid, voice
3/5	id.	beli	beli	low
3/5	id.	mukli	mukli	high
3/5	id.	bliska	bliska	dor, son
3/5	id.	bodro	bodro	ant, cor, delrel, distr, low, strid
3/5	id.	sitni	sitni	high
4/5	id.	milo	milo	high
4/5	id.	grub	grub	ant, cor, distr, strid
4/5	id.	vaz̩no	vaz̩no	tense
4/5	id.	bogati	bogati	low
5/5	non-id.	mil	mio	high
5/5	non-id.	vaz̩n	vaz̩an	ant
5/5	id.	rapav	rapav	cont, labdent
5/5	id.	blag	blag	low
5/5	id.	blaga	blaga	low
Manually imposed phonological feature combinations: R2L				
1/5	non-id.	dobr	dobar	ant, cor, distr, lab, strid
1/5	non-id.	obl	obao	lab
1/5	id.	mukla	mukla	low

Fold	Type	UR	Gold SR	Failed ϕ
1/5	id.	uska	uska	high
2/5	non-id.	mukl	mukao	dor, high, low, voice
2/5	non-id.	ustal	ustao	front, low
2/5	id.	zelena	zelena	low
2/5	id.	zeleni	zeleni	low
2/5	id.	vaz̨ni	vaz̨ni	low
3/5	non-id.	blizk	blizak	ant, cor, delrel, distr, dor, high, low, son, strid, voice
3/5	non-id.	uzk	uzak	voice
3/5	id.	duboko	duboko	dor, low, nasal
3/5	id.	kratko	kratko	ant, cor, delrel, distr, high, strid, voice
3/5	id.	bogat	bogat	dor, son
3/5	id.	uski	uski	ant, cor, delrel, distr, high, strid, voice
3/5	id.	beli	beli	low
3/5	id.	mukli	mukli	high
3/5	id.	bliska	bliska	dor, son
3/5	id.	bodro	bodro	ant, cor, delrel, distr, low, strid
3/5	id.	sitni	sitni	high

Fold	Type	UR	Gold SR	Failed ϕ
4/5	id.	milo	milo	high
4/5	id.	grub	grub	ant, cor, distr, strid
5/5	non-id.	mil	mio	high
5/5	non-id.	vaz̩n	vaz̩an	ant
5/5	id.	rapav	rapav	labdent
5/5	id.	obla	obla	low
5/5	id.	podla	podla	low

TABLE C.19: Serbo-Croatian held-out forms that were mispredicted in the 5-fold cross-validation experiment.

C.6 Polish: CHILDES

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
1/10	non-id.	kɔnikɔv	kɔnikuf	voice	80
1/10	id.	mamusiami	mamusiami	voice	80
1/10	id.	vid͡ziawa	vid͡ziawa	voice	80
1/10	id.	ɔpɔviɛd͡ziawam	ɔpɔviɛd͡ziawam	voice	80
1/10	id.	t͡ʂɨtat͡siɛ	t͡ʂɨtat͡siɛ	voice	80
1/10	id.	znalɛʑt͡ɕ	znalɛʑt͡ɕ	voice	80
1/10	id.	znajdɔ̃	znajdɔ̃	voice	80
1/10	id.	znalazwɛm	znalazwɛm	voice	80
1/10	id.	wuzʲɛk	wuzʲɛk	voice	80
1/10	id.	usiɛdli	usiɛdli	voice	80
1/10	id.	pzʲɛt͡ʂɨtawɨ	pzʲɛt͡ʂɨtawɨ	voice	80
1/10	id.	nɔt͡s	nɔt͡s	voice	80
1/10	id.	kzʲɨt͡ʂɛ̃	kzʲɨt͡ʂɛ̃	voice	80
1/10	id.	lɛzʲaw	lɛzʲaw	high	110
1/10	id.	lɛzʲɨʂ	lɛzʲɨʂ	tense	110
1/10	id.	lɛt͡sɨʂ	lɛt͡sɨʂ	high, tense	110
1/10	id.	lɛt͡siawam	lɛt͡siawam	high, tense	110
1/10	id.	zawɔzʲɨwaɕ	zawɔzʲɨwaɕ	high, tense	110
1/10	id.	t͡siɛkavɨx	t͡siɛkavɨx	high	110
1/10	id.	kɔtku	kɔtku	high	110
1/10	id.	kulkɛ̃	kulkɛ̃	tense	110
1/10	id.	lampax	lampax	high	110
1/10	id.	ɕviɛt͡siɛ	ɕviɛt͡siɛ	tense	110
1/10	id.	kupujɔ̃	kupujɔ̃	high	110

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
1/10	id.	kupɔvawa	kupɔvawa	tense	110
1/10	id.	pɔdam	pɔdam	high	110
1/10	id.	nastɛ̃pnɛgɔ	nastɛ̃pnɛgɔ	tense	110
1/10	id.	t͡sisɛkavɨm	t͡sisɛkavɨm	high, tense	1410
1/10	id.	zdjɛ̃wɨ	zdjɛ̃wɨ	tense	1410
2/10	id.	ɕpiɔ̃t͡sɛgɔ	ɕpiɔ̃t͡sɛgɔ	voice	80
2/10	id.	jɛd͡zisi	jɛd͡zisi	voice	80
2/10	id.	dɔbra	dɔbra	voice	80
2/10	id.	pisaliɕt͡si	pisaliɕt͡si	voice	80
2/10	id.	pwakawa	pwakawa	voice	80
2/10	id.	ubrawɛm	ubrawɛm	voice	80
2/10	id.	usiadwa	usiadwa	voice	80
2/10	id.	vilki	vilki	voice	80
2/10	id.	ruzɲɨmi	ruzɲɨmi	voice	80
2/10	id.	ɔtvɔzʲɨwɨ	ɔtvɔzʲɨwɨ	voice	80
2/10	id.	ut͡sisɛkali	ut͡sisɛkali	voice	80
2/10	id.	svɛj	svɛj	high	110
2/10	id.	misisɛm	misisɛm	tense	110
2/10	id.	vɨjmisɛʂ	vɨjmisɛʂ	high	110
2/10	id.	nisɛbɛm	nisɛbɛm	tense	110
2/10	id.	d͡ziawɨ	d͡ziawɨ	high	110
2/10	id.	rat͡sja	rat͡sja	tense	110
2/10	id.	pravd͡zivɔ̃	pravd͡zivɔ̃	high	110
2/10	id.	t͡sisɛʂɨli	t͡sisɛʂɨli	tense	110
3/10	id.	sisɛd͡zisɛli	sisɛd͡zisɛli	voice	80
3/10	id.	d͡zisɛvt͡ʂɨnki	d͡zisɛvt͡ʂɨnki	voice	80

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
3/10	id.	t͡ʂɛrvɔnɨm	t͡ʂɛrvɔnɨm	voice	80
3/10	id.	swɨʂɨ	swɨʂɨ	voice	80
3/10	id.	umɨwa	umɨwa	voice	80
3/10	id.	biwɛm	biwɛm	voice	80
3/10	id.	zɔstali	zɔstali	voice	80
3/10	id.	skɔɲt͡ʂɨwaɕ	skɔɲt͡ʂɨwaɕ	voice	80
3/10	id.	usiɔ̃d͡zɪɛmɨ	usiɔ̃d͡zɪɛmɨ	voice	80
3/10	id.	prat͡sɔvawa	prat͡sɔvawa	voice	80
3/10	non-id.	samɔxɔdɔv	samɔxɔduf	tense	110
3/10	non-id.	ɔbiadɔv	ɔbiaduf	tense	110
3/10	non-id.	swɔv	swuf	high, tense	110
3/10	id.	samɔxɔdɨ	samɔxɔdɨ	high	110
3/10	id.	stɔit͡sɪɛ	stɔit͡sɪɛ	high	110
3/10	id.	vɪɛlkɪɛgɔ	vɪɛlkɪɛgɔ	tense, voice	110
3/10	id.	zapɔmnɪɛt͡ɕ	zapɔmnɪɛt͡ɕ	high	110
3/10	id.	t͡sɪɛkavɛ	t͡sɪɛkavɛ	tense	110
3/10	id.	budujt͡sɪɛ	budujt͡sɪɛ	high, tense	110
3/10	id.	zbɪɛrawam	zbɪɛrawam	high	110
3/10	id.	zbudujɛ	zbudujɛ	tense	110
3/10	non-id.	ɕnɪɛg	ɕnɪɛk	high	240
3/10	non-id.	svɔj	svuj	high, tense	1480
4/10	id.	musiawɨbɨɕmɨ	musiawɨbɨɕmɨ	voice	80
4/10	id.	prɔsit͡ɕ	prɔsit͡ɕ	voice	80
4/10	id.	pzʲɨnɪɛɕlɪɕt͡sɪɛ	pzʲɨnɪɛɕlɪɕt͡sɪɛ	voice	80
4/10	id.	gzʐɛt͡ʂnɨx	gzʐɛt͡ʂnɨx	voice	80
4/10	id.	sxɔvam	sxɔvam	voice	80

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
4/10	id.	znalazwɛɕ	znalazwɛɕ	voice	80
4/10	id.	nautʂʲimʲ	nautʂʲimʲ	voice	80
4/10	id.	ɕpiɛvaj	ɕpiɛvaj	voice	80
4/10	id.	biawɛgɔ	biawɛgɔ	voice	80
4/10	id.	ɕlitʂnɛ	ɕlitʂnɛ	voice	80
4/10	non-id.	tatusiɔv	tatusiuf	high, tense	110
4/10	non-id.	panɔv	panuf	high, tense	110
4/10	non-id.	tʂwɔviɛkɔv	tʂwɔviɛkuf	tense	110
4/10	non-id.	krakɔv	krakuf	high	110
4/10	id.	patʐawaɕ	patʐawaɕ	high	110
4/10	id.	gurʲ	gurʲ	high, tense	110
4/10	id.	umʲtʲ	umʲtʲ	high, tense	110
4/10	id.	biwʲ	biwʲ	high, tense	110
4/10	id.	pwʲvatɕ	pwʲvatɕ	high	110
4/10	id.	pwʲvaliɕmʲ	pwʲvaliɕmʲ	tense	110
4/10	id.	fajniɛ	fajniɛ	tense	110
4/10	id.	vrutsiwɛɕ	vrutsiwɛɕ	high	110
4/10	id.	ɕrɔdku	ɕrɔdku	tense	110
5/10	id.	druga	druga	voice	80
5/10	id.	pʐʲniɔswʲɕmʲ	pʐʲniɔswʲɕmʲ	voice	80
5/10	id.	swuxawa	swuxawa	voice	80
5/10	id.	niɛmɛ	niɛmɛ	voice	80
5/10	id.	stawʲ	stawʲ	voice	80
5/10	id.	utʂʲ	utʂʲ	voice	80
5/10	id.	umʲwam	umʲwam	voice	80
5/10	id.	uxɔ	uxɔ	voice	80

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
5/10	id.	zawɔzᶣɔnɨmi	zawɔzᶣɔnɨmi	voice	80
5/10	non-id.	t͡ʂasɔv	t͡ʂasuf	high, tense	110
5/10	id.	fajnɪɛjʂɨ	fajnɪɛjʂɨ	high	110
5/10	id.	najfajnɪɛjʂa	najfajnɪɛjʂa	tense	110
5/10	id.	skɔɲt͡ʂɨʂ	skɔɲt͡ʂɨʂ	high	110
5/10	id.	stɔwɛm	stɔwɛm	tense	110
5/10	id.	pɔmɔgɔ̃	pɔmɔgɔ̃	high	110
5/10	id.	pɔmɔgwaɕ	pɔmɔgwaɕ	tense	110
5/10	id.	kupɔvawɨ	kupɔvawɨ	high	110
5/10	id.	papiɛrɨ	papiɛrɨ	tense	110
5/10	id.	kruvkɛ̃	kruvkɛ̃	high	110
5/10	id.	gwuvkɛ̃	gwuvkɛ̃	tense	110
5/10	id.	nɔsit͡ɕ	nɔsit͡ɕ	high	110
5/10	non-id.	nɔg	nuk	high, tense	230
5/10	id.	znamɨ	znamɨ	high, tense	2210
6/10	id.	musiawbɨɕ	musiawbɨɕ	voice	80
6/10	id.	musiawaɕ	musiawaɕ	voice	80
6/10	id.	prɔʂɔ̃	prɔʂɔ̃	voice	80
6/10	id.	mnɪɛjʂɔ̃	mnɪɛjʂɔ̃	voice	80
6/10	id.	pwat͡ʂɔ̃	pwat͡ʂɔ̃	voice	80
6/10	id.	znajd͡zɪɛmɨ	znajd͡zɪɛmɨ	voice	80
6/10	id.	ɕpiɛvawam	ɕpiɛvawam	voice	80
6/10	id.	biali	biali	voice	80
6/10	id.	pɔmɔzᶣɛt͡siɛ	pɔmɔzᶣɛt͡siɛ	voice	80
6/10	id.	vilku	vilku	voice	80
6/10	non-id.	dɔmɔv	dɔmuf	high, tense	110

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
6/10	id.	swɨʂawam	swɨʂawam	high	110
6/10	id.	swɨʂawɛɕ	swɨʂawɛɕ	tense	110
6/10	id.	zamknɛ̃wɔ	zamknɛ̃wɔ	high	110
6/10	id.	lɛsiɛ	lɛsiɛ	tense	110
6/10	id.	pzɻɛvrut͡sili	pzɻɛvrut͡sili	high	110
6/10	id.	radia	radia	tense	110
6/10	id.	tramvajɛ	tramvajɛ	high	110
6/10	id.	vɨjɛ̃waɕ	vɨjɛ̃waɕ	tense	110
6/10	id.	d͡zid͡ziuɕ	d͡zid͡ziuɕ	high	110
6/10	id.	buduj	buduj	tense	110
6/10	id.	kɔ̃pawɔ	kɔ̃pawɔ	high	110
6/10	id.	vɨlawɛm	vɨlawɛm	tense	110
6/10	id.	t͡siɛʂɨw	t͡siɛʂɨw	high	110
6/10	id.	siadajɔ̃	siadajɔ̃	tense	110
6/10	id.	zat͡ʂɛ̃wɨɕmɨ	zat͡ʂɛ̃wɨɕmɨ	high	110
6/10	id.	pɔspzɻ̃tawa	pɔspzɻ̃tawa	tense	110
6/10	id.	pɔdɔbawɨ	pɔdɔbawɨ	high	230
6/10	id.	brawa	brawa	tense	970
7/10	id.	spawɛm	spawɛm	voice	80
7/10	id.	najgzɻɛt͡ʂniɛjʂa	najgzɻɛt͡ʂniɛjʂa	voice	80
7/10	id.	graliɕt͡siɛ	graliɕt͡siɛ	voice	80
7/10	id.	samɔxɔd͡ziɛ	samɔxɔd͡ziɛ	voice	80
7/10	id.	ʂukaliɕmɨ	ʂukaliɕmɨ	voice	80
7/10	id.	ʂɨbkɔ	ʂɨbkɔ	voice	80
7/10	id.	vɨjd͡ziɛt͡siɛ	vɨjd͡ziɛt͡siɛ	voice	80
7/10	id.	ɔpɔviadajɔ̃	ɔpɔviadajɔ̃	voice	80

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
7/10	id.	skɔɲtʂɨtɕ	skɔɲtʂɨtɕ	voice	80
7/10	id.	zabrawɨ	zabrawɨ	voice	80
7/10	non-id.	bəl	bul	high	120
7/10	non-id.	pəməzˌ	pəmuʂ	high	120
7/10	non-id.	vilkɔv	vilkuf	high	120
7/10	non-id.	lasɔv	lasuf	high	120
7/10	non-id.	psɔv	psuf	high, tense	120
7/10	non-id.	krɔv	kruf	high	120
7/10	id.	napiʂɛ	napiʂɛ	tense	120
7/10	id.	usiɔ̃dzie	usiɔ̃dzie	tense	120
7/10	id.	xwɔpiɛts	xwɔpiɛts	tense	120
7/10	id.	ptaʂɛk	ptaʂɛk	tense	120
7/10	id.	zawɔzʲiwɛɕ	zawɔzʲiwɛɕ	tense	120
7/10	id.	tsiɛʂɨwɨ	tsiɛʂɨwɨ	tense	120
7/10	id.	najgɔrʂɛ	najgɔrʂɛ	high	120
7/10	id.	zwamatɕ	zwamatɕ	high	240
8/10	non-id.	xwɔptʂɨkɔv	xwɔptʂɨkuf	voice	80
8/10	id.	baviwɨ	baviwɨ	voice	80
8/10	id.	tzɛma	tzɛma	voice	80
8/10	id.	miɛʂkali	miɛʂkali	voice	80
8/10	id.	vwɔzʲiwam	vwɔzʲiwam	voice	80
8/10	id.	zɛpsuwɛɕ	zɛpsuwɛɕ	voice	80
8/10	id.	biwa	biwa	voice	80
8/10	id.	ɔkna	ɔkna	voice	80
8/10	id.	ɔbrazkiɛm	ɔbrazkiɛm	voice	80
8/10	id.	ɔddawɛɕ	ɔddawɛɕ	voice	80

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
8/10	id.	ɔglɔ̃damɨ	ɔglɔ̃damɨ	voice	80
8/10	id.	utsiɛkamɨ	utsiɛkamɨ	voice	80
8/10	id.	gzɻɛtʂnɛgɔ	gzɻɛtʂnɛgɔ	tense	230
8/10	id.	ɔknami	ɔknami	tense	230
8/10	id.	kɔt	kɔt	tense	230
8/10	id.	piɛ̃knɨm	piɛ̃knɨm	tense	230
8/10	id.	dwugiɛgɔ	dwugiɛgɔ	tense	230
8/10	id.	kɔlatsjɔ̃	kɔlatsjɔ̃	tense	230
8/10	id.	patzɻ̃ts	patzɻ̃ts	tense	230
8/10	id.	pɔlitʂiwa	pɔlitʂiwa	high	230
8/10	id.	pɔdanɔ̃	pɔdanɔ̃	tense	230
8/10	non-id.	rɔb	rup	high, tense, voice	290
8/10	id.	gzɻɛtʂnɨm	gzɻɛtʂnɨm	high	290
8/10	id.	pzɻjɛxawɨ	pzɻjɛxawɨ	high	290
8/10	id.	pɨtaw	pɨtaw	high	290
8/10	id.	uʂu	uʂu	high	290
8/10	id.	pratsɔvali	pratsɔvali	high	290
8/10	id.	niuniu	niuniu	high	290
8/10	id.	najgɔrʂim	najgɔrʂim	high	290
9/10	non-id.	viɛdz	viɛts	voice	110
9/10	non-id.	dziadkɔv	dziadkuf	tense	110
9/10	non-id.	vwɔzɻ	vwuʂ	tense	110
9/10	non-id.	dɔmkɔv	dɔmkuf	tense	110
9/10	non-id.	vwɔsɔv	vwɔsuf	tense	110
9/10	id.	ɕpiɔ̃tsɛ	ɕpiɔ̃tsɛ	voice	110

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
9/10	id.	siɛd͡ziawɨɕt͡siɛ	siɛd͡ziawɨɕt͡siɛ	voice	110
9/10	id.	t͡ʂwɔviɛki	t͡ʂwɔviɛki	high	110
9/10	id.	ksiɔ̃ʐ̩kami	ksiɔ̃ʐ̩kami	voice	110
9/10	id.	rɨsujɔ̃	rɨsujɔ̃	high	110
9/10	id.	vwɔʐɨwaɕ	vwɔʐɨwaɕ	voice	110
9/10	id.	ksiɔ̃ʐɛt͡ʂki	ksiɔ̃ʐɛt͡ʂki	high	110
9/10	id.	piɛ̃kniɛjʂɨ	piɛ̃kniɛjʂɨ	high	110
9/10	id.	vkwadaj	vkwadaj	voice	110
9/10	id.	vkwadaliɕmɨ	vkwadaliɕmɨ	voice	110
9/10	id.	wuʐɛt͡ʂku	wuʐɛt͡ʂku	voice	110
9/10	id.	drɔga	drɔga	tense	110
9/10	id.	vɨbrat͡ɕ	vɨbrat͡ɕ	high, tense	110
9/10	id.	vɨbiɛʐ̩mɨ	vɨbiɛʐ̩mɨ	high, tense	110
9/10	id.	drɔgi	drɔgi	high	230
9/10	non-id.	stɔw	stuw	tense	1420
9/10	id.	skɔɲt͡ʂɨt͡siɛ	skɔɲt͡ʂɨt͡siɛ	high	1420
10/10	id.	pɔviɛd͡ziɛliɕmɨ	pɔviɛd͡ziɛliɕmɨ	voice	80
10/10	id.	dawam	dawam	voice	80
10/10	id.	viɛ̃kʂɨx	viɛ̃kʂɨx	voice	80
10/10	id.	nɔgax	nɔgax	voice	80
10/10	id.	piwa	piwa	voice	80
10/10	id.	rɔ̃t͡ʂɛk	rɔ̃t͡ʂɛk	voice	80
10/10	id.	napisat͡ɕ	napisat͡ɕ	voice	80
10/10	id.	jɛʑd͡zit͡ɕ	jɛʑd͡zit͡ɕ	voice	80
10/10	id.	umɨtɛ	umɨtɛ	voice	80
10/10	id.	strɔn	strɔn	voice	80

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
10/10	id.	ubranɛ	ubranɛ	voice	80
10/10	non-id.	mɔv	muf	high, tense	110
10/10	id.	usiadwaɕ	usiadwaɕ	high	110
10/10	id.	pɔla	pɔla	tense	110
10/10	id.	paluʂku	paluʂku	high	110
10/10	id.	zamkniɛ̃tɨx	zamkniɛ̃tɨx	tense	110
10/10	id.	xwɔpt͡ʂɨk	xwɔpt͡ʂɨk	high	110
10/10	id.	xɔrɨ	xɔrɨ	tense	110
10/10	id.	vxɔd͡zi	vxɔd͡zi	tense	110
10/10	id.	vɨʐut͡sɔnɛ	vɨʐut͡sɔnɛ	high	110
10/10	id.	pzɛʂkad͡zat͡sɨɛ	pzɛʂkad͡zat͡sɨɛ	tense	110
10/10	non-id.	xɔd͡ʑ	xut͡ɕ	high, tense	980
10/10	non-id.	pɔkɔj	pɔkuj	tense	980
10/10	id.	mɨjɔ̃	mɨjɔ̃	high	980
10/10	id.	ɕrɔdkiɛm	ɕrɔdkiɛm	high	980

TABLE C.20: Polish held-out forms that were mispredicted at one or more points in the batch-size-10 incremental R2L identity-learner experiment.

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
1/10	non-id.	kɔnikɔv	kɔnikuf	voice	100
1/10	id.	mamusiami	mamusiami	voice	100
1/10	id.	vid͡ziawa	vid͡ziawa	voice	100
1/10	id.	ɔpɔviɛd͡ziawam	ɔpɔviɛd͡ziawam	voice	100
1/10	id.	t͡ʂɨtat͡siɛ	t͡ʂɨtat͡siɛ	voice	100
1/10	id.	znalɛʑt͡ɕ	znalɛʑt͡ɕ	voice	100
1/10	id.	znajdɔ̃	znajdɔ̃	voice	100
1/10	id.	znalazwɛm	znalazwɛm	voice	100
1/10	id.	wuzʐɛk	wuzʐɛk	voice	100
1/10	id.	usiɛdli	usiɛdli	voice	100
1/10	id.	pzʐɛt͡ʂɨtawɨ	pzʐɛt͡ʂɨtawɨ	voice	100
1/10	id.	nɔt͡s	nɔt͡s	voice	100
1/10	id.	kzʐɨt͡ʂɛ̃	kzʐɨt͡ʂɛ̃	voice	100
1/10	id.	lɛzʐaw	lɛzʐaw	high	200
1/10	id.	lɛzʐɨʂ	lɛzʐɨʂ	tense	200
1/10	id.	lɛt͡siʂ	lɛt͡siʂ	high, tense	200
1/10	id.	lɛt͡siawam	lɛt͡siawam	high, tense	200
1/10	id.	zawɔzʐɨwaɕ	zawɔzʐɨwaɕ	high, tense	200
1/10	id.	t͡siɛkavɨx	t͡siɛkavɨx	high	200
1/10	id.	kɔtku	kɔtku	high	200
1/10	id.	kulkɛ̃	kulkɛ̃	tense	200
1/10	id.	lampax	lampax	high	200
1/10	id.	ɕviɛt͡siɛ	ɕviɛt͡siɛ	tense	200
1/10	id.	kupujɔ̃	kupujɔ̃	high	200
1/10	id.	kupɔvawa	kupɔvawa	tense	200
1/10	id.	pɔdam	pɔdam	high	200

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
1/10	id.	nastɛ̃pnɛgɔ	nastɛ̃pnɛgɔ	tense	200
1/10	id.	t͡sisɛkavɨm	t͡sisɛkavɨm	high, tense	1500
1/10	id.	zdjɛ̃wɨ	zdjɛ̃wɨ	tense	1500
2/10	id.	ɕpiɔ̃t͡sɛgɔ	ɕpiɔ̃t͡sɛgɔ	voice	100
2/10	id.	jɛd͡zisɛt͡sisɛ	jɛd͡zisɛt͡sisɛ	voice	100
2/10	id.	dɔbra	dɔbra	voice	100
2/10	id.	pisaliɕt͡sisɛ	pisaliɕt͡sisɛ	voice	100
2/10	id.	pwakawa	pwakawa	voice	100
2/10	id.	ubrawɛm	ubrawɛm	voice	100
2/10	id.	usiadwa	usiadwa	voice	100
2/10	id.	vilki	vilki	voice	100
2/10	id.	ruzɲɨmi	ruzɲɨmi	voice	100
2/10	id.	ɔtvɔzʲɨwɨ	ɔtvɔzʲɨwɨ	voice	100
2/10	id.	ut͡sisɛkali	ut͡sisɛkali	voice	100
2/10	id.	svɛj	svɛj	high	200
2/10	id.	misisɛm	misisɛm	tense	200
2/10	id.	vɨjmisɛʂ	vɨjmisɛʂ	high	200
2/10	id.	nisɛbɛm	nisɛbɛm	tense	200
2/10	id.	d͡ziawɨ	d͡ziawɨ	high	200
2/10	id.	rat͡sja	rat͡sja	tense	200
2/10	id.	pravd͡zivɔ̃	pravd͡zivɔ̃	high	200
2/10	id.	t͡sisɛʂɨli	t͡sisɛʂɨli	tense	200
3/10	id.	sisɛd͡zisɛli	sisɛd͡zisɛli	voice	100
3/10	id.	d͡zisɛvt͡ʂɨnki	d͡zisɛvt͡ʂɨnki	voice	100
3/10	id.	t͡ʂɛrvɔnɨm	t͡ʂɛrvɔnɨm	voice	100
3/10	id.	swɨʂɨ	swɨʂɨ	voice	100

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
3/10	id.	umɨwa	umɨwa	voice	100
3/10	id.	biwɛm	biwɛm	voice	100
3/10	id.	zɔstali	zɔstali	voice	100
3/10	id.	skɔɲt͡ʂɨwaɕ	skɔɲt͡ʂɨwaɕ	voice	100
3/10	id.	usiɔ̃d͡zismɨ	usiɔ̃d͡zismɨ	voice	100
3/10	id.	prat͡sɔvawa	prat͡sɔvawa	voice	100
3/10	non-id.	samɔxɔdɔv	samɔxɔduf	tense	200
3/10	non-id.	ɔbiadɔv	ɔbiaduf	tense	200
3/10	non-id.	swɔv	swuf	high, tense	200
3/10	id.	samɔxɔdɨ	samɔxɔdɨ	high	200
3/10	id.	stɔit͡sis	stɔit͡sis	high	200
3/10	id.	vislkisgɔ	vislkisgɔ	tense, voice	200
3/10	id.	zapɔmnist͡ɕ	zapɔmnist͡ɕ	high	200
3/10	id.	t͡sisɛkavɛ	t͡sisɛkavɛ	tense	200
3/10	id.	budujt͡sis	budujt͡sis	high, tense	200
3/10	id.	zbisɛrawam	zbisɛrawam	high	200
3/10	id.	zbudujɛ	zbudujɛ	tense	200
3/10	non-id.	ɕnisg	ɕnisk	high	300
3/10	non-id.	svɔj	svuj	high, tense	1500
4/10	id.	musiawɨbɨɕmɨ	musiawɨbɨɕmɨ	voice	100
4/10	id.	prɔsit͡ɕ	prɔsit͡ɕ	voice	100
4/10	id.	pʑɨnisɕlit͡ɕsis	pʑɨnisɕlit͡ɕsis	voice	100
4/10	id.	gzʑɛt͡ʂnɨx	gzʑɛt͡ʂnɨx	voice	100
4/10	id.	sxɔvam	sxɔvam	voice	100
4/10	id.	znalazwɛɕ	znalazwɛɕ	voice	100
4/10	id.	naut͡ʂɨmɨ	naut͡ʂɨmɨ	voice	100

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
4/10	id.	ɕpiɛvaj	ɕpiɛvaj	voice	100
4/10	id.	biawɛgɔ	biawɛgɔ	voice	100
4/10	id.	ɕlit͡ʂnɛ	ɕlit͡ʂnɛ	voice	100
4/10	non-id.	tatusiɔv	tatusiuf	high, tense	200
4/10	non-id.	panɔv	panuf	high, tense	200
4/10	non-id.	t͡ʂwɔviɛkɔv	t͡ʂwɔviɛkuf	tense	200
4/10	non-id.	krakɔv	krakuf	high	200
4/10	id.	patʐawaɕ	patʐawaɕ	high	200
4/10	id.	gurɨ	gurɨ	high, tense	200
4/10	id.	umɨtɨ	umɨtɨ	high, tense	200
4/10	id.	biwɨ	biwɨ	high, tense	200
4/10	id.	pwɨvat͡ɕ	pwɨvat͡ɕ	high	200
4/10	id.	pwɨvaliɕmɨ	pwɨvaliɕmɨ	tense	200
4/10	id.	fajniɛ	fajniɛ	tense	200
4/10	id.	vrut͡siwɛɕ	vrut͡siwɛɕ	high	200
4/10	id.	ɕrɔdku	ɕrɔdku	tense	200
5/10	id.	druga	druga	voice	100
5/10	id.	pʐɨniɔswɨɕmɨ	pʐɨniɔswɨɕmɨ	voice	100
5/10	id.	swuxawa	swuxawa	voice	100
5/10	id.	niɛmɛ	niɛmɛ	voice	100
5/10	id.	stawɨ	stawɨ	voice	100
5/10	id.	ut͡ʂɨ	ut͡ʂɨ	voice	100
5/10	id.	umɨwam	umɨwam	voice	100
5/10	id.	uxɔ	uxɔ	voice	100
5/10	id.	zawɔzʲɲɨmi	zawɔzʲɲɨmi	voice	100
5/10	non-id.	t͡ʂasɔv	t͡ʂasuf	high, tense	200

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
5/10	id.	fajnɪɛjʂɨ	fajnɪɛjʂɨ	high	200
5/10	id.	najfajnɪɛjʂa	najfajnɪɛjʂa	tense	200
5/10	id.	skɔɲt͡ʂɨʂ	skɔɲt͡ʂɨʂ	high	200
5/10	id.	stɔwɛm	stɔwɛm	tense	200
5/10	id.	pɔmɔgɔ̃	pɔmɔgɔ̃	high	200
5/10	id.	pɔmɔgwaɕ	pɔmɔgwaɕ	tense	200
5/10	id.	kupɔvawɨ	kupɔvawɨ	high	200
5/10	id.	papiɛrɨ	papiɛrɨ	tense	200
5/10	id.	kruvkɛ̃	kruvkɛ̃	high	200
5/10	id.	gwuvkɛ̃	gwuvkɛ̃	tense	200
5/10	id.	nɔsit͡ɕ	nɔsit͡ɕ	high	200
5/10	non-id.	nɔg	nuk	high, tense	300
5/10	id.	znamɨ	znamɨ	high, tense	2300
6/10	id.	musiawbɨɕ	musiawbɨɕ	voice	100
6/10	id.	musiawaɕ	musiawaɕ	voice	100
6/10	id.	prɔʂɔ̃	prɔʂɔ̃	voice	100
6/10	id.	mnɪɛjʂɔ̃	mnɪɛjʂɔ̃	voice	100
6/10	id.	pwat͡ʂɔ̃	pwat͡ʂɔ̃	voice	100
6/10	id.	znajd͡zɪɛmɨ	znajd͡zɪɛmɨ	voice	100
6/10	id.	ɕpiɛvawam	ɕpiɛvawam	voice	100
6/10	id.	biali	biali	voice	100
6/10	id.	pɔmɔzʐɛt͡siɛ	pɔmɔzʐɛt͡siɛ	voice	100
6/10	id.	vilku	vilku	voice	100
6/10	non-id.	dɔmɔv	dɔmuf	high, tense	200
6/10	id.	swɨʂawam	swɨʂawam	high	200
6/10	id.	swɨʂawɛɕ	swɨʂawɛɕ	tense	200

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
6/10	id.	zamknɛ̃wɔ	zamknɛ̃wɔ	high	200
6/10	id.	lɛsiɛ	lɛsiɛ	tense	200
6/10	id.	pzɻɛvrut͡sili	pzɻɛvrut͡sili	high	200
6/10	id.	radia	radia	tense	200
6/10	id.	tramvajɛ	tramvajɛ	high	200
6/10	id.	vɨjɛ̃waɕ	vɨjɛ̃waɕ	tense	200
6/10	id.	d͡zid͡ziuɕ	d͡zid͡ziuɕ	high	200
6/10	id.	buduj	buduj	tense	200
6/10	id.	kɔ̃pawɔ	kɔ̃pawɔ	high	200
6/10	id.	vɨlawɛm	vɨlawɛm	tense	200
6/10	id.	t͡siɛʂɨw	t͡siɛʂɨw	high	200
6/10	id.	siadajɔ̃	siadajɔ̃	tense	200
6/10	id.	zat͡ʂɛ̃wɨɕmɨ	zat͡ʂɛ̃wɨɕmɨ	high	200
6/10	id.	pɔspzɻ̃tawa	pɔspzɻ̃tawa	tense	200
6/10	id.	pɔdɔbawɨ	pɔdɔbawɨ	high	300
6/10	id.	brawa	brawa	tense	1000
7/10	id.	spawɛm	spawɛm	voice	100
7/10	id.	najgzɻɛt͡ʂniɛjʂa	najgzɻɛt͡ʂniɛjʂa	voice	100
7/10	id.	graliɕt͡siɛ	graliɕt͡siɛ	voice	100
7/10	id.	samɔxɔd͡ziɛ	samɔxɔd͡ziɛ	voice	100
7/10	id.	ʂukaliɕmɨ	ʂukaliɕmɨ	voice	100
7/10	id.	ʂɨbkɔ	ʂɨbkɔ	voice	100
7/10	id.	vɨjd͡ziɛt͡siɛ	vɨjd͡ziɛt͡siɛ	voice	100
7/10	id.	ɔpɔviadajɔ̃	ɔpɔviadajɔ̃	voice	100
7/10	id.	skɔɲt͡ʂɨt͡ɕ	skɔɲt͡ʂɨt͡ɕ	voice	100
7/10	id.	zabrawɨ	zabrawɨ	voice	100

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
7/10	non-id.	bɔl	bul	high	200
7/10	non-id.	pɔmɔzˌ	pɔmuˌs	high	200
7/10	non-id.	vilkɔv	vilkuf	high	200
7/10	non-id.	lasɔv	lasuf	high	200
7/10	non-id.	psɔv	psuf	high, tense	200
7/10	non-id.	krɔv	kruf	high	200
7/10	id.	napiˌsɛ	napiˌsɛ	tense	200
7/10	id.	usiɔ̃d͡ziɛ	usiɔ̃d͡ziɛ	tense	200
7/10	id.	xwɔpiɛt͡s	xwɔpiɛt͡s	tense	200
7/10	id.	ptaˌsɛk	ptaˌsɛk	tense	200
7/10	id.	zawɔzˌɨwɛɕ	zawɔzˌɨwɛɕ	tense	200
7/10	id.	t͡siɛˌsɨwɨ	t͡siɛˌsɨwɨ	tense	200
7/10	id.	najgɔrˌsɛ	najgɔrˌsɛ	high	200
7/10	id.	zwamat͡ɕ	zwamat͡ɕ	high	300
8/10	non-id.	xwɔpt͡ˌsɨkɔv	xwɔpt͡ˌsɨkuf	voice	100
8/10	id.	baviwɨ	baviwɨ	voice	100
8/10	id.	tzˌɛma	tzˌɛma	voice	100
8/10	id.	miɛˌskali	miɛˌskali	voice	100
8/10	id.	vwɔzˌɨwam	vwɔzˌɨwam	voice	100
8/10	id.	zɛpsuwɛɕ	zɛpsuwɛɕ	voice	100
8/10	id.	biwa	biwa	voice	100
8/10	id.	ɔkna	ɔkna	voice	100
8/10	id.	ɔbrazkiɛm	ɔbrazkiɛm	voice	100
8/10	id.	ɔddawɛɕ	ɔddawɛɕ	voice	100
8/10	id.	ɔglɔ̃damɨ	ɔglɔ̃damɨ	voice	100
8/10	id.	ut͡siɛkamɨ	ut͡siɛkamɨ	voice	100

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
8/10	non-id.	rɔb	rup	high, tense, voice	300
8/10	id.	gzʐɛt͡ʂnɨm	gzʐɛt͡ʂnɨm	high	300
8/10	id.	gzʐɛt͡ʂnɛgɔ	gzʐɛt͡ʂnɛgɔ	tense	300
8/10	id.	pzʲɨjɛxawɨ	pzʲɨjɛxawɨ	high	300
8/10	id.	ɔknami	ɔknami	tense	300
8/10	id.	pɨtaw	pɨtaw	high	300
8/10	id.	kɔt	kɔt	tense	300
8/10	id.	uʂu	uʂu	high	300
8/10	id.	piɛ̃knɨm	piɛ̃knɨm	tense	300
8/10	id.	prat͡sɔvali	prat͡sɔvali	high	300
8/10	id.	dwugiɛgɔ	dwugiɛgɔ	tense	300
8/10	id.	niuniu	niuniu	high	300
8/10	id.	kɔlat͡sjɔ̃	kɔlat͡sjɔ̃	tense	300
8/10	id.	najgɔrʂɨm	najgɔrʂɨm	high	300
8/10	id.	patzɔ̃t͡s	patzɔ̃t͡s	tense	300
8/10	id.	pɔlit͡ʂɨwa	pɔlit͡ʂɨwa	high	300
8/10	id.	pɔdanɔ̃	pɔdanɔ̃	tense	300
9/10	non-id.	viɛd͡z	viɛt͡s	voice	200
9/10	non-id.	d͡ziadkɔv	d͡ziadkuf	tense	200
9/10	non-id.	vwɔzˌ	vwuʂ	tense	200
9/10	non-id.	dɔmkɔv	dɔmkuf	tense	200
9/10	non-id.	vwɔsɔv	vwɔsuf	tense	200
9/10	id.	ɕpiɔ̃t͡sɛ	ɕpiɔ̃t͡sɛ	voice	200
9/10	id.	siɛd͡ziawɨɕt͡siɛ	siɛd͡ziawɨɕt͡siɛ	voice	200
9/10	id.	t͡ʂwɔviɛki	t͡ʂwɔviɛki	high	200

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
9/10	id.	ksiɔ̃zʲkami	ksiɔ̃zʲkami	voice	200
9/10	id.	rɨsujɔ̃	rɨsujɔ̃	high	200
9/10	id.	vwɔzʲɨwaɕ	vwɔzʲɨwaɕ	voice	200
9/10	id.	ksiɔ̃zʲɛt͡ʂki	ksiɔ̃zʲɛt͡ʂki	high	200
9/10	id.	piɛ̃kniɛjʂɨ	piɛ̃kniɛjʂɨ	high	200
9/10	id.	vkwadaj	vkwadaj	voice	200
9/10	id.	vkwadaliɕmɨ	vkwadaliɕmɨ	voice	200
9/10	id.	wuzʲɛt͡ʂku	wuzʲɛt͡ʂku	voice	200
9/10	id.	drɔga	drɔga	tense	200
9/10	id.	vɨbrat͡ɕ	vɨbrat͡ɕ	high, tense	200
9/10	id.	vɨbiɛzʲmɨ	vɨbiɛzʲmɨ	high, tense	200
9/10	id.	drɔgi	drɔgi	high	300
9/10	non-id.	stɔw	stuw	tense	1500
9/10	id.	skɔɲt͡ʂɨt͡siɛ	skɔɲt͡ʂɨt͡siɛ	high	1500
10/10	id.	pɔviɛd͡ziɛliɕmɨ	pɔviɛd͡ziɛliɕmɨ	voice	100
10/10	id.	dawam	dawam	voice	100
10/10	id.	viɛ̃kʂɨx	viɛ̃kʂɨx	voice	100
10/10	id.	nɔgax	nɔgax	voice	100
10/10	id.	piwa	piwa	voice	100
10/10	id.	rɔ̃t͡ʂɛk	rɔ̃t͡ʂɛk	voice	100
10/10	id.	napisat͡ɕ	napisat͡ɕ	voice	100
10/10	id.	jɛʑd͡zit͡ɕ	jɛʑd͡zit͡ɕ	voice	100
10/10	id.	umɨtɛ	umɨtɛ	voice	100
10/10	id.	strɔn	strɔn	voice	100
10/10	id.	ubranɛ	ubranɛ	voice	100
10/10	non-id.	mɔv	muf	high, tense	200

Fold	Type	UR	Gold SR	Failed ϕ	Passed at
10/10	id.	usiadwaɕ	usiadwaɕ	high	200
10/10	id.	pɔla	pɔla	tense	200
10/10	id.	paluʂku	paluʂku	high	200
10/10	id.	zamkniɛ̃tɨx	zamkniɛ̃tɨx	tense	200
10/10	id.	xwɔpt͡ʂɨk	xwɔpt͡ʂɨk	high	200
10/10	id.	xɔrɨ	xɔrɨ	tense	200
10/10	id.	vxɔd͡zi	vxɔd͡zi	tense	200
10/10	id.	vɨzut͡sɔnɛ	vɨzut͡sɔnɛ	high	200
10/10	id.	pzɛʂkad͡zat͡sɨɛ	pzɛʂkad͡zat͡sɨɛ	tense	200
10/10	non-id.	xɔd͡ʑ	xut͡ɕ	high, tense	1000
10/10	non-id.	pɔkɔj	pɔkuj	tense	1000
10/10	id.	mɨjɔ̃	mɨjɔ̃	high	1000
10/10	id.	ɕrɔdkiɛm	ɕrɔdkiɛm	high	1000

TABLE C.21: Polish held-out forms that were mispredicted at one or more points in the incremental R2L identity-learner experiment.