Tier-based modeling of gradience and distance-based decay in phonological processes

Kevin McMullin
University of Ottawa
kevin.mcmullin@uottawa.ca

Phillip Burness
University of Ottawa
pburn036@uottawa.ca

Abstract

Current computational approaches to long-distance phonological processes use string-to-string function classes that operate over phonological tiers, but these are necessarily deterministic devices and are thus limited to enforcing categorical application. We show that probabilistic relations that act like a tier-based function (aside from being non-deterministic) perform well as models of gradient long-distance processes. In particular, they offer a cognitively plausible characterization of *distance-based decay* (Zymet, 2015) with other desirable properties, exemplified by two case studies. The first, examining rounding dissimilation in Malagasy, demonstrates that tier-based models of decay can be made sensitive to phonetic similarity in interesting ways. The second, examining Hungarian backness harmony, demonstrates that tier-based models of decay can handle scenarios where a process is obligatory at short distances but vanishingly unlikely at increasing distances.

1 Introduction

Taking inspiration from foundational results in *Autosegmental Phonology* (e.g., Goldsmith 1976), recent computational work has shown that long-distance phonological processes can be fruitfully modelled using string-to-string function classes that operate according to a relativized notion of strict locality. These classes include the Tier-based Strictly Local (TSL) functions explored by Burness and McMullin (2019), Hao and Andersson (2019), Hao and Bowers (2019), and Andersson et al. (2020), as well as the Multi-tiered Strictly Local (MTSL) functions (Burness and McMullin, 2020). While these functions have offered valuable insights into the computational characteristics of non-local phonology, they are limited in that they assume every input has exactly one output. Consequently, these functions can only describe either mandatory application or mandatory non-application of a process. Real language data is, however, not always this clean; many phonological processes apply *optionally*, and long-distance processes are no exception to this fact. This paper will explore how the requirement of determinism can be relaxed in order to describe the probabilistic application of a process, while still maintaining the advantages of (tier-based) strict locality. In particular, by augmenting the tier-based structures with duplicate transitions for non-tier elements, we are able to model phonological processes with a well-known property of *distance-based decay*, wherein the probability that a long-distance process applies will exponentially diminish as more and more transparent segments intervene between the trigger and target (Zymet, 2015).

The paper is structured as follows. First, Section 2 provides the necessary background on tier-based functions and their automata-theoretic characterization. Then, Section 3 looks at some optional long-distance patterns and shows how strategically adding transitions to a TSL or MTSL FST and weighting them can describe the desired probabilistic distribution of output forms for a given input. After that, Section 4 considers distance-based decay, and proposes that weighted transducers built according to a TSL or MTSL template can derive distance-based decay in a cognitively plausible manner. Section 5 concludes.

2 Categorical tier-based functions

We begin this section with a modicum of notation and definitions An *alphabet* is a set of elements from which strings can be built. The concatenation of two strings u and v is written as $u \cdot v$, although this is shortened to uv when context permits. Given an alphabet Σ, we write Σ^* to denote the set of all strings of any length (including 0) that can be

Proceedings of the 17th Meeting on the Mathematics of Language, pages 50–63, Montpellier (online),
December 13, 2021. © 2021 Association for Computational Linguistics

constructed using Σ. Here and throughout, we use λ to denote the unique *empty string*, which has a length of 0 and satisfies $\lambda \cdot w = w \cdot \lambda = w$. Given an alphabet Σ of input elements and an alphabet Γ of output elements, a (partial) *string-to-string function* is a mapping from Σ^* to Γ^* where each $w \in \Sigma^*$ is paired with (at most) one string in Γ^*.

We will mainly demonstrate and discuss tier-based functions (and their probabilistic variants) with reference to the *finite-state transducers* (FSTs) that compute them. A (one-way) FST produces an output string incrementally by reading an input string one element at a time in a single direction. Such a machine consists of a finite set of states (which can be thought of as a primitive sort of memory) and a finite set of transitions between these states (which are the machine's instructions for what to write at each step). The machine begins in a designated initial state, and traverses a path through the state space by following transitions in response to the input that it reads. Each state is given a (potentially empty) *final string* which is appended to the output when the machine lands in that state after consuming the whole input string. Figure 1 presents a visual diagram of an FST. States are represented using circles and the initial state is marked with an unlabeled incoming arrow. Transitions are represented with labelled arrows between states; a label 'a:b' is an instruction to take that transition when reading 'a' from the input and write 'b' to the output. Final strings are shown underneath the state label, formatted like a transition label for the special end-marker $\ltimes$. The transducer in Figure 1 operates over the input alphabet {a, b}, transforming all odd-numbered positions to 'a', transforming all even-numbered positions to 'b', and appending 'b' to the end if it runs out of input after writing 'a' (i.e., if it ends in the state labelled '1'). For example it maps /bab/ to [abab] and maps /aabbab/ to [ababab].

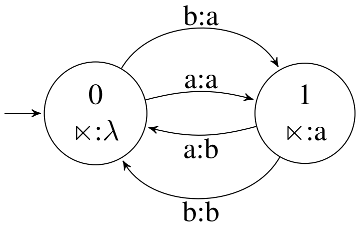

Figure 1: A simple finite-state transducer

The tier-based functions expanded upon in this paper are themselves extensions of the Strictly Local functions (Chandlee, 2014; Chandlee et al., 2014, 2015, 2018; Chandlee and Heinz, 2018). A Strictly k-Local (SL$_k$) FST operates according to a memory window with a fixed and finite maximum size k, the state labels acting as a record of this window's contents. When the window pays attention to the input string we say that the function is Input Strictly k-Local (ISL$_k$) and the label of the currently occupied state always corresponds to the most recent (up to) $k-1$ elements read. Similarly, when the window pays attention to the output string we say that the function is Output Strictly k-Local (OSL$_k$) and the label of the currently occupied state always corresponds to the most recent (up to) $k-1$ elements written. Input Tier-based Strictly k-Local (ITSL$_k$) and Output Tier-based Strictly k-Local (OTSL$_k$) FSTs operate exactly like their ISL$_k$ and OSL$_k$ cousins except that only a subset of the relevant alphabet (the function's *tier*) is allowed to occupy space in the memory window. Restricting the transducers attention in this manner permits the modelling of non-local processes where the distance between trigger and target can be arbitrarily large.

To demonstrate, consider the process of regressive sibilant harmony in Slovenian. The Slovenian process is optional, though we will assume for the purposes of this section that it is categorical, postponing a discussion of its optionality to Section 3. The Slovenian pattern causes a sibilant to become [−anterior] if it is followed at any distance by another [−anterior] sibilant unless a coronal stop intervenes (Jurgec, 2011, pp. 329-333). Examples of successful sibilant harmony are provided in (1a-b) and examples of blocked sibilant harmony are provided in (1c-d) with the second singular suffix acting as the potential trigger in each case.

(1) Slovenian sibilant harmony, blocking by coronal stops (Jurgec, 2011, pp. 330-331)

a.	/spi-ʃ/	[ʃpi-ʃ]	'sleep-2SG'
b.	/pozabi-ʃ/	[poʒabi-ʃ]	'forget-2SG'
c.	/stoji-ʃ/	[stoji-ʃ]	'stand-2SG'
d.	/zida-ʃ/	[zida-ʃ]	'build-2SG'

The transducer in Figure 2 shows what an idealized and mandatory version of the Slovenian pattern would look like as an OTSL$_2$ function operating relative to the tier {s, ʃ, z, ʒ, t, d}. Since the process is regressive, the machine reads input strings from right to left. State labels are enclosed in square brackets to highlight the fact that this transducer tracks the output string. To save on space, states with the same behaviour are collapsed into a single circle with multiple labels and transitions that share an origin and a destination are collapsed into a single arrow with multiple labels. Note that transitions labelled ‘*:*’ represent an arbitrary non-tier segment mapping faithfully to itself. An input [+anterior] sibilant (i.e., /s/ or /z/) will map faithfully to itself if no tier-elements have been produced thus far, if the most recently produced tier element was a coronal stop, or if the most recently produced tier element was another [+anterior] sibilant. On the other hand, if the most recently produced tier element was a [−anterior] sibilant (i.e., [ʃ] or [ʒ]), an input [+anterior] sibilant will instead palatalise to become its [−anterior] equivalent. Palatalization will happen no matter how many non-tier elements (i.e., non-sibilants other than [t] or [d]) intervene between the [−anterior] trigger and the [+ anterior] target, because producing such an element never causes a change of state in this machine.

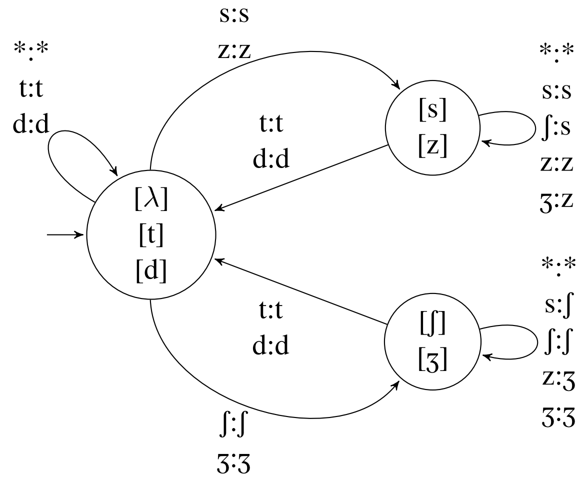

Figure 2: An OTSL$_2$ transducer that produces mandatory sibilant harmony

ITSL$_k$ and OTSL$_k$ functions are equipped with just one tier, which works well for many cases, but some patterns require multiple memory windows that each track a different tier. By allowing for multiple tiers in this way we delve into the class of Multi-Tiered Strictly k-Local functions (Burness and McMullin, 2020). All of the MTSL$_k$ transducers that will appear in this paper adhere to a restriction which Burness and McMullin (2020) call *target-specification*. The restriction states that (i) each input element is associated with a set of tiers that on their own can fully determine what the element is mapped to on a given step and (ii) this *target-specified* set of tiers must form a strict superset-subset hierarchy. Target-specified MTSL functions essentially track multiple, related sources of information when deciding how to process a particular input element. Tiers that are not part of a input element’s specified set are ignored when reading that input element, since they provide either irrelevant or redundant information.

3 Probabilistic variants

In the previous section, we mentioned that the Slovenian process of sibilant harmony was optional. When the transducer in Figure 2 reads an input like /pozabiʃ/ from right to left, it will be in the [−anterior] state as it goes to read /z/, and the corresponding transition will enforce harmony. We want, however, to have the possibility of faithfully producing [z] for /z/ while in the [−anterior] state, since harmony is optional in Slovenian (Jurgec, 2011). We can create the possibility of optional faithfulness by adding transitions from the [−anterior] state to the [+anterior] state labelled ‘s:s’ and ‘z:z’ and transitions from the [+anterior] state to the [−anterior] state labelled ‘ʃ:ʃ’ and ‘ʒ:ʒ’. Figure 3 shows the resulting transducer, which aside from the added non-determinism, exhibits all the required behaviour of an OTSL$_2$ transducer (i.e., all transitions still land in the state corresponding to the most recently written tier element). For clarity, the harmony-enforcing transitions are shown as dotted lines and the harmony-ignoring (faithful) transitions are shown as dashed lines.

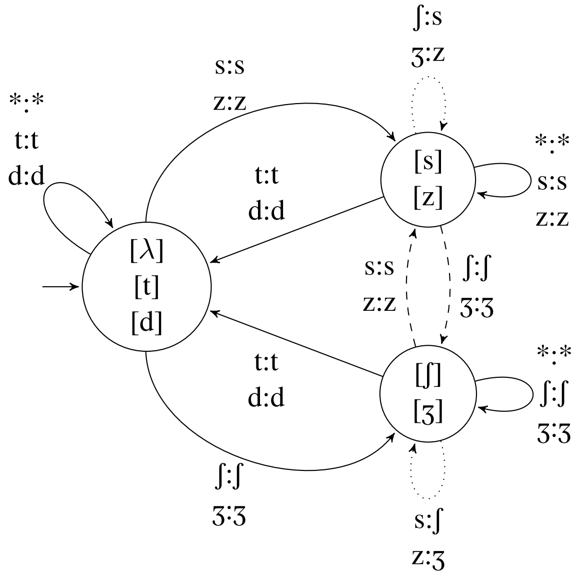

Figure 3: A quasi-OTSL$_2$ transducer that produces optional sibilant harmony

Now we have two transitions out of the [−anterior] state for the input /z/, one that enforces harmony and one that enforces faithfulness. Assuming that we choose randomly between the two available transitions, the /z/ in /pozabiʃ/ then has a 50% chance of harmonizing with the nearby [ʃ]. It is very important that the new faithful transition leads to the [+anterior] state rather than looping back to the [−anterior] state. This is because, while the new faithful transition does not produce harmony, it nonetheless produces a tier element. By ensuring that any additional transitions all lead to the state associated with the most recently produced tier element, we maximally preserve the intuitions of the TSL functions, even though we are abandoning determinism and thus no longer meet the definition of a TSL function. We instead have a *quasi*-TSL *relation*, where the set of possible outputs at a given step is directly determined by the most recently produced tier element.

Of course, we may want to achieve a rate of harmony higher than 50% while still allowing for the possibility of faithfulness. To do so, we can assign a numerical weight to each transition in the machine (Vidal et al., 2005a). When more than one transition could be followed at a given step in the derivation, the probability that we choose a given member from that set of transitions is proportional to its share of the summed weights of the set. For ease of interpretation, we assume that the weight of a transition is equal to the probability that it is followed, meaning that given a state q and an input symbol a, the weights of all transitions leaving q for the input a must add up to 1. Suppose now that the harmony-ignoring (dashed) transitions are weighted 0.2, the harmony-enforcing (dotted) transitions are weighted 0.8, and the remaining transitions are weighted 1. The input /sapozabiʃ/ has three possible outcomes whose probabilities sum to 1: a fully faithful candidate [sapozabiʃ], a single harmony candidate [sapoʒabiʃ], and a full harmony candidate [ʃapoʒabiʃ]. The fully faithful candidate has a probability of $0.2 * 1 = 0.2$ since the probability of the /z/ remaining faithful is 0.2, and if it does so, the /s/ is guaranteed to be faithful. The remaining output probabilities can be calculated in a similar manner: the single harmony candidate has a probability of $0.8 * 0.2 = 0.16$, and the full harmony candidate has a probability of $0.8 * 0.8 = 0.64$. More generally, a weighted transducer set up in the above manner produces a conditional distribution over a finite set of output strings for each possible input string. The number of output possibilities as well as their shape and share of the probability can of course change from input to input and from transducer to transducer, but the distributions so-defined are crucially related to and dictated by tier-based strict locality.

A particularly interesting case of optionality in a long-distance process comes from Bukusu. In this language, underlying /l/ becomes output [r] if the nearest leftward surface liquid is [r] (de Blois, 1975; Odden, 1994; Hansson, 2010). The pattern of liquid harmony affects the applicative suffix /-ila/, exemplified by the data in (2). The suffix's underlying /l/ surfaces faithfully when the root contains no liquids as in (2a) or when the only liquids in the root are all instances of /l/ as in (2b). When the base contains an /r/, though, the liquid in the applicative suffix alternates to obey harmony. This happens across a single vowel as in (2c) and at further distances as in (2d).

(2) Bukusu liquid harmony (Odden, 1994)

a.	xam-ila	'milk-APPL'
b.	lim-ila	'cultivate-APPL'
c.	kar-ira	'twist-APPL'
d.	rum-ira	'send-APPL'

Importantly, harmony is obligatory across a single vowel (i.e., in transvocalic contexts) but becomes optional at further distances (Hansson, 2010). For example, /ruk-ila/ 'plait-APPL' may surface as [ruk-ila] without harmony or as [ruk-ira] with harmony. Another long-distance pattern that is cited as being obligatory in transvocalic contexts but optional at further distances would be the sibilant harmony in Kinyarwanda (Kimenyi, 1979; Coupez, 1980; Hansson, 2010; Walker and Mpiranya, 2006; Walker et al., 2008). Such a transvocalic / beyond-transvocalic dichotomy is not possible to describe using a probabilistic quasi-OTSL$_2$ transducer as we did for Slovenian sibilant harmony above, but *is* possible to describe using a probabilistic quasi-OMTSL$_2$ transducer.

Consider the transducer in Figure 4, where 'V' stands for an arbitrary vowel and 'C' stands for an arbitrary non-liquid consonant. Ignoring the dashed transition for now (but including the dotted transitions), this machine represents a target-specified OMTSL$_2$ function that computes a fully obligatory version of the Bukusu pattern over a tier of liquid consonants $A = \{\text{r}, \text{l}\}$ and a tier of all consonants $B = \{\text{r}, \text{l}, \text{C}\}$. The left and right symbol of each state label correspond respectively to the suffix on A (i.e., the most recently produced liquid consonant) and B (i.e., the most recently produced consonant). Reaching the [r, r] state can be interpreted to mean that the most recent consonant we have seen is an [r] (since it is on both the liquid and consonantal tiers). Compare this to being in the [r, C] state, which means that the most recent consonant we have seen is a non-liquid, and that this consonant is preceded by an [r]. In a transducer that does not contain the dashed transition, both of these states enforce the harmonic /l/ → [r] change, as indicated by the dotted transitions. However, by adding the dashed transition, harmony becomes optional just in those cases where [r] is the most recently produced liquid but not the most recently produced consonant. In a language like Bukusu with mostly open CV syllables, these two states more-or-less reflect the difference between a transvocalic and beyond-transvocalic distance from the most recent liquid consonant. In particular it is the superset-subset relationship imposed onto the tierset by target specification (Burness and McMullin, 2020) that allows us to have harmony be obligatory across 0 non-liquid consonants and be optional across 1+ non-liquid consonants.

An important question arises when we model optional processes using probabilistic transducers. Namely, how do we determine the transition weights that best reflect the target pattern? This type of optimization problem is well-studied in the literature on Probabilistic Finite-state Acceptors (PFAs) which are exactly like probabilistic transducers except that rather than taking an input string and producing an output string, they take an input string and return a value reflecting the input's well-formedness. The Slovenian and Bukusu transducers above can be reinterpreted as acceptors if we think of their transition labels as atomic elements of an alphabet and rewrite input-output pairs as a string of such "transducer actions". Conveniently, reinterpreting the Slovenian and Bukusu transducers in this manner makes them deterministic since, while a given input string can follow potentially multiple paths through the transducers to produce different outputs, a given input-output pair can only be achieved by following a single, specific path through the transducers. Finding the transition weights for a given deterministic PFA that maximise the probability of a set of training data has a well-known, simple, and efficient solution. For each transition,[1] we calculate the number of times it was followed when reading the sample and divide this number by the total number of times its origin state was visited when reading the sample (Vidal et al., 2005a,b; de la Higuera, 2010). One small modification is needed for our purposes since the weights resulting from the above method will describe a single distribution over input-output pairs, rather than a separate distribution over output strings for each input. This is because the weights of all transitions out of a given state will sum to 1, whereas we want all transitions out of a given state *for a given input element* to sum to 1. To remedy this, we can normalize the transducer by taking each combination of state and input symbol, adding together the weights of all transitions leaving that state for that input element, then dividing each of the implicated transition weights by this sum.

4 Distance-based decay

The analyses of the Slovenian and Bukusu cases above are relatively simplistic in that the probability with which the process applies remains constant. In many cases, however, we see that the probability of application is inversely correlated to the distance between trigger and target. This phenomenon is known as *distance-based decay* (Zymet, 2015) and can be observed in Malagasy vowel rounding dissimilation (Zymet, 2015), Hungarian backness vowel harmony (Hayes and Londe, 2006; Hayes et al., 2009), Latin liquid dissimilation (Zymet, 2015), and Navajo sibilant harmony (Martin, 2005), among others. Current descriptions of the phenomenon are couched within stochastic constraint-based frameworks like Noisy Harmonic Grammar (Coetzee and Pater, 2011) and Maximum Entropy grammar (Goldwater and Johnson, 2003; Hayes and Wilson, 2008). These descriptions propose that the weight of a process-enforcing constraint is scaled down proportionally to the distance between trigger and target (Kimper, 2011; Zymet, 2015). Distant trigger-target pairs incur smaller penalties than more local pairs, and as a result, the process applies at a lower probability in the distant pair than in the more local pair (Kimper, 2011; Zymet, 2015). Focusing on Malagasy and Hungarian, we will show how distance-based decay can equally be captured through minor modifications

[1] The final strings associated to states are treated as transitions for the purposes of this optimization, effectively acting as a transitions that lead to a dedicated "stopping" state with no associated string of its own.

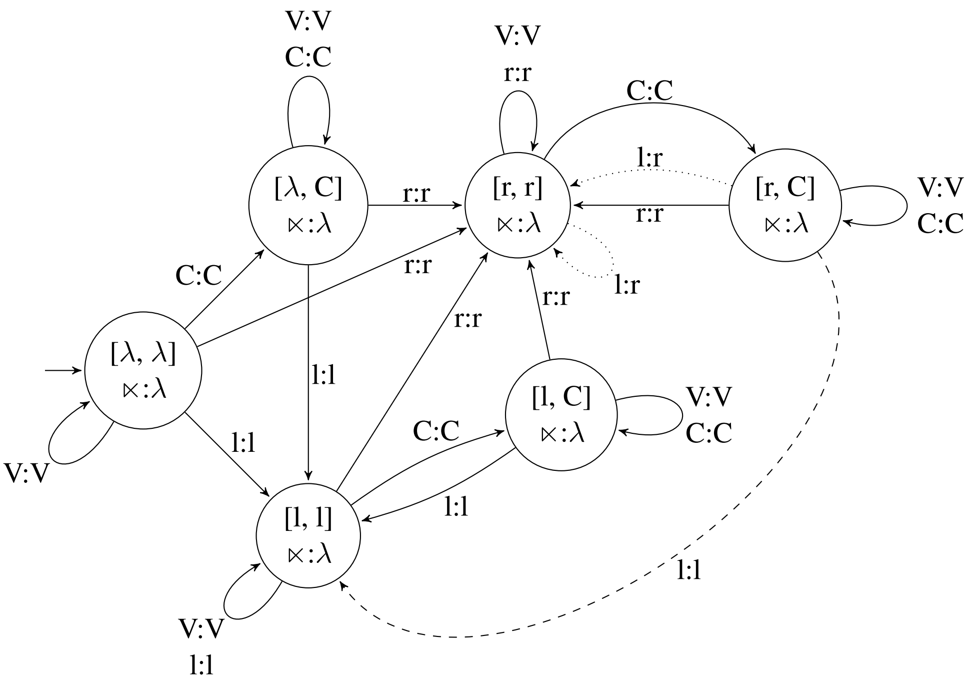

Figure 4: A quasi-OMTSL$_2$ transducer that computes Bukusu liquid harmony

to a TSL or MTSL transducer.

4.1 Malagasy

Malagasy has a process of vowel rounding dissimilation whereby the passive imperfective suffix /-u/ becomes [-i] when preceded by an [u], as can be seen in /babu-u/ → [babu-i] 'plunder-PASS.IMP'. Front vowels are opaque to the process as can be seen with /turi-u/ → [turi-u] 'preach-PASS.IMP' and /ure-u/ → [ure-u] 'massage-PASS.IMP'. In contrast, the vowel /a/ is transparent to dissimilation, as can be seen with /gurabah-u/ → [gurabah-i] 'splutter-PASS.IMP'. If we ignore its dashed transition (discussed further below), the OTSL$_2$ transducer in Figure 5 captures a mandatory version of the Malagasy pattern just described. Dissimilation specifically affects the passive imperative suffix rather than /u/ in general (Zymet, 2020), so for convenience we assume that this transducer only ever reads verb stems and adds the appropriate suffix allomorph upon reaching the end of the base.

Malagasy dissimilation is not categorical, however, and exhibits distance-based decay. According to Zymet's (2015) survey of de la Beaujardière's (2004) online Malagasy dictionary, the probability of dissimilation is 0.99 (989/993) when the trigger and target are in adjacent syllables, 0.51 (201/397) when they are separated by one transparent syllable, 0.13 (4/32) when they are separated by two transparent syllables, and 0 (0/4) when they are separated by three transparent syllables. Due to the language's mostly open-syllable nature, the number of transparent syllables corresponds with how many transparent vowels fall between the trigger and target, and the probability of dissimilation is roughly $1/2^x$, where x represents the number of

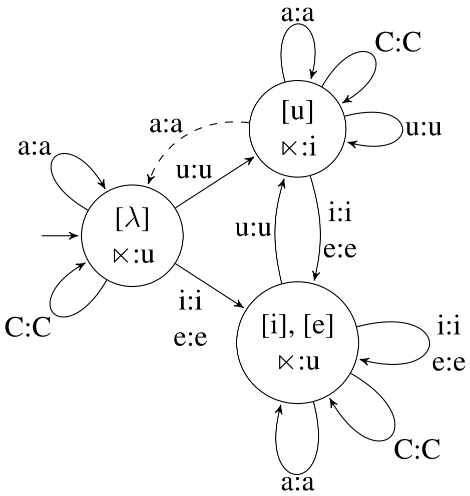

Figure 5: A quasi-OTSL$_2$ transducer that produces Malagasy dissimilation

intervening transparent vowels. This opens an interesting route to deriving the distance-based decay using the structure of the transducer in Figure 5: whenever the transducer reads /a/ and produces [a] while in the [u] state, it has a roughly 50% chance to "forget" that it previously produced an instance of [u]. In this case, the transducer will follow the dashed transition which returns to the [λ] state instead of looping back to the [u] state, and will consequently fail to dissimilate the passive imperative suffix. As more transparent vowels are encountered while in the [u] state, the machine is exponentially less likely to remember that it encountered a dissimilation trigger, giving us the negative exponential curve in the probability of dissimilation. A cognitive interpretation of forgetful transitions would be that the memory of the most recent tier element decays over time.

Two related questions arise when modelling distance-based decay by augmenting a TSL transducer with forgetfulness parameters. First, how do we decide which forgetful transitions should be added to the TSL transducer? Any transition that fails to produce an element from the tier can presumably be given a forgetful version, but including too many or too few of these could negatively impact the accuracy of our model. Second, given a fixed set of forgetful transitions, how do we determine their optimal weights? We answer the latter question first, since its solution will be considered when approaching the former question.

Recall from Section 3 that to optimize the weights of a deterministic acceptor, it is sufficient to read through the provided sample once and count the number of times that each transition is followed. Unfortunately, even after reinterpreting a forgetful quasi-TSL transducer as an acceptor, it is still non-deterministic. Given an input-output pair we can generally tell whether forgetting did or did not occur, but we cannot tell exactly where the forgetting took place when there is a sequence of more than one transparent element. Because we cannot always know the exact path that an input-output pair followed through the machine, we cannot accurately count the number of times each transition get traversed when the sample is read. It is, however, possible to estimate these counts given the machine's current transition weights using what are called forward and backward probabilities. Consider the element x in the string $w = u \cdot x \cdot v$. For a transition labeled x leaving state q and landing in state q' we can calculate the probability that we are in state q after having read u (the forward probability) and the probability that we produce v when starting in state q' (the backward probability).[2] Multiplying the current weight of a given transition by its forward and backward probability and then dividing by the probability of the whole string gives us the probability that we actually traversed the transition on that reading step (de la Higuera, 2010, pp. 362-363).

By using estimated traversal probabilities as our traversal counts, we can calibrate the weights of a non-deterministic acceptor using the same division operations as for a deterministic acceptor. If we cycle through the estimation and calibration processes just described, the parameter weights will get adjusted by smaller and smaller amounts until they converge. This is known as the Baum-Welch algorithm,[3] originally developed by Baum et al. (1970) and Baum (1972). It is a type of maximum likelihood estimation (MLE) that, metaphorically, climbs the "hill" of sample probabilities by adjusting the available parameter values, and stops when it reaches a peak and cannot increase the sample probability any further.

The algorithm is guaranteed to converge on such an optimum, but non-deterministic machines can have multiple optima in addition to the global optimum, and the algorithm may get trapped in one of these (Vidal et al., 2005b; de la Higuera, 2010). Returning to the hill metaphor, there can be multiple peaks of varying heights and we want the algorithm to find the highest one, but it cannot tell whether the peak it reaches is actually the highest, it simply stops once it finds *any* peak. The only guaranteed way around this is to try several times with different starting values, and then pick the result that gives the best probability, in the hope that the chosen iteration found the global optimum (de la Higuera, 2010, p. 323). Luckily, this was not a serious issue during the tests described further below. Whenever a transducer needed optimizing, the optimization process was run several times with random initializations of the transducer's transition weights, and each machine always achieved the same approximate log-likelihood no matter its initialization, suggesting that (at least in these cases) there were no local optima in which the optimizer

[2] Chapter 5 of de la Higuera (2010) shows how to efficiently calculate forward and backward probabilities.

[3] See chapter 17 of de la Higuera (2010) for a more thorough presentation.

could get trapped. The results reported for each machine below are relative to the optimization that achieved the best log-likelihood.

Moving on to the question of which forgetful transitions to include, we could take the stance that we want only the forgetful transitions that significantly affect model performance. So long as the set of forgetful transitions in one optimized transducer are a strict superset of the forgetful transitions in another optimized transducer, it is in theory possible to perform a log-likelihood ratio test to assess whether the additional transitions significantly improve model performance. Given a calibrated transducer, we calculate its log-likelihood by running the training sample through it. For each input-output pair (x, y) we calculate the probability that the machine produces y given x, which is equal to the sum of the probability of all paths through the machine that produce y given x. This might seem difficult to do efficiently since the number of possible paths through a non-deterministic transducer is in the worst-case exponentially proportional to the length of the input string, but we can bypass this issue by calculating forward probabilities (which takes just one pass through the string) and summing over those instead (de la Higuera, 2010, pp. 90-92). If we then take the log of each pair's probability, adding them up gives us the model's log-likelihood. One way to find the best set of forgetful transitions, then, would be to take a forwards selection approach. Starting with no forgetful parameters, we iteratively add the one that would contribute the most until we cannot significantly improve our model any more.

To test the effectiveness of the FST decay model, we created custom Python code that implements the Baum-Welch algorithm and log-likelihood calculation procedures described above, then ran it against the Malagasy data from Zymet (2015). Calibrating the base model affects only the probability that dissimilation occurs while in the [u] state, and the optimal value of 83.73% gives a log-likelihood of -633.30. Most additional forgetful transitions significantly improved model fit on their own,[4] but 'a' had by far the strongest contribution, increasing log-likelihood all the way to -315.34 ($\chi^2_1 = 635.93, p = 2.57 \times 10^{-140}$). The next highest contribution came from 'l', which increased log-likelihood to -609.67 ($\chi^2_1 = 47.27$, $p = 6.2 \times 10^{-12}$). A second round of tests using the 'a' model only found two additional forgetful transitions to be significant: these were 'dʒ' ($\chi^2_1 = 3.85, p = 0.050$) and 'z' ($\chi^2_1 = 3.93, p = 0.048$). This might seem odd considering how most were highly significant on the first round of tests. Looking at the largely CV syllable structure of Malagasy, though, the presence of an intervening 'a' heavily implies the presence of an intervening consonant. Significant contributions from the lone consonantal parameters may thus have been indirect inheritances from instances of 'a'. Because forgetful 'dʒ' and 'z' transitions are just barely significant given a threshold of $p < 0.05$, we opted not to include either and stop further testing, leaving us with just a forgetful 'a' transition. One reason that 'a' may have near-exclusive entitlement to a forgetful transition is its high similarity to the tier elements, all of which are vowels. Encountering a non-tier element that is highly similar to elements on the tier would intuitively interfere with the maintenance of a tier suffix in long-term memory, although we leave the confirmation of this hypothesis for future research.

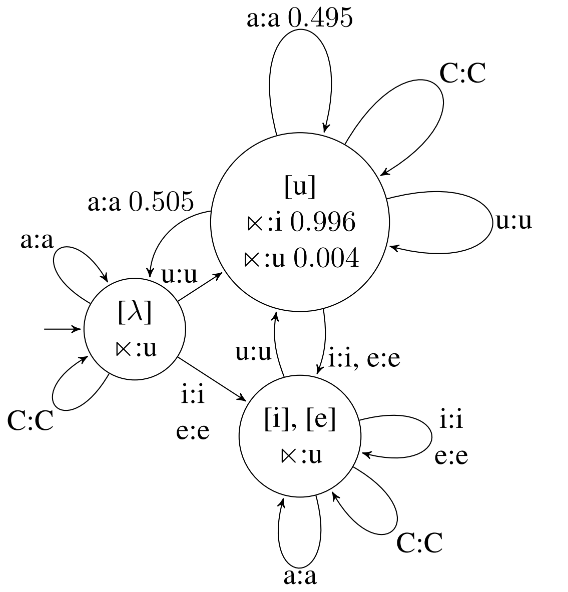

Figure 6: An optimized quasi-OTSL$_2$ transducer for Malagasy

The optimized Malagasy transducer is shown in Figure 6; all transitions without a displayed weight have a weight of 1. Earlier we mentioned that, as reported by (Zymet, 2015), the probability of dissimilation is $0.996(989/993)$ when the trigger and

[4] Only 'v' ($\chi^2_1 = 3.58, p = 0.06$), 't' ($\chi^2_1 = 2.66, p = 0.1$), 'f' ($\chi^2_1 = 0.49, p = 0.48$) and 'h' ($\chi^2_1 = 0, p = 1$) did not. The last case is particularly interesting in that the optimal weight for a lone forgetful 'h' transition was 0, equivalent to the absence of such a transition.

target are in adjacent syllables, $0.506(201/397)$ when they are separated by one transparent syllable, $0.125(4/32)$ when they are separated by two transparent syllables, and $0.00(0/4)$ when they are separated by three transparent syllables. A single forgetful transition for [a] pretty faithfully reproduces the probability of adjacent dissimilation (0.996) and dissimilation across one transparent syllable $(0.996 * 0.495 = 0.493)$, but modestly overestimates the probability of dissimilation across two intervening syllables $(0.996 * 0.495^2 = 0.244)$ and three intervening syllables $(0.996 * 0.495^3 = 0.121)$. Zymet's (2015) constraint-based model more closely reproduces the latter two probabilities, but this may be an instance of overfitting, considering how few forms in the corpus contain 2+ intervening syllables. In any case, the model with a forgetful [a] transition drastically outperforms the base model, which predicts dissimilation with a probability of 0.837 at any distance.

4.2 Hungarian

Including forgetfulness parameters into a single-tiered transducer is sufficient for the Malagasy case, but not all cases of distance-based decay are so easy. Take for instance the backness vowel harmony in Hungarian, to which [i], [e], and [ɛ] are transparent (Hayes and Londe, 2006; Hayes et al., 2009; Kimper, 2011; Ozburn, 2019). While it is generally true that a higher number of transparent vowels between trigger and target will exponentially diminish the probability of harmony, there is an important exception: harmony remains nearly obligatory across a single transparent vowel (Hayes and Londe, 2006; Hayes et al., 2009; Kimper, 2011; Ozburn, 2019). For example, the [ɔ] in [pɔpiːr] 'paper' always triggers the back variant of the dative suffix ([pɔpiːr-nɔk] 'paper-DAT') since it is followed by only one transparent vowel, but the [ɔ] in [ɔspirin] 'aspirin' only optionally triggers the back variant since it is followed by two transparent vowels ([ɔspirin-nɔk] ∼ [ɔspirin-nɛk] 'aspirin-DAT').

This is impossible to model using a single-tiered transducer, even with forgetful transitions. To see why, consider the transducer fragment in Figure 7, which determines the appropriate allomorph of the dative suffix /-nEk/ for bases containing only high vowels.[5] The underspecified suffix vowel /E/ must harmonize while in either the /u/ or /y/ state, and defaults to front while in the [λ] state. The vowel /i/ is transparent to harmony and so each of these states has a looping transition labelled 'i:i'. We could try modelling the distance-based decay using the forgetful transitions marked with dashed lines, but these do not distinguish between having one transparent vowel and having two or more transparent vowels between trigger and target. We want forgetfulness to begin applying only in the latter case, but there is no way to set such a threshold on the required number of transparent segments in a single-tiered transducer. In such a transducer, a transparent segment either always or never has the opportunity to cause forgetfulness.

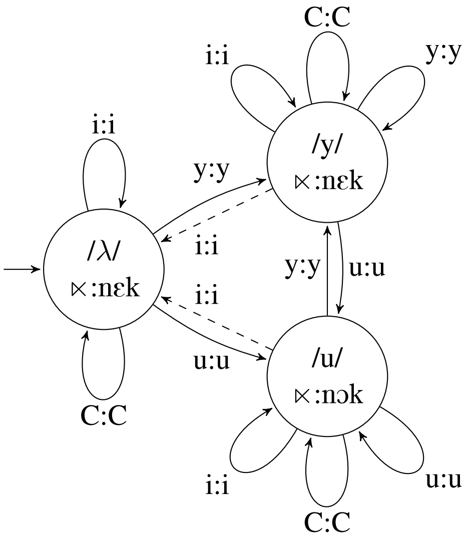

Figure 7: A quasi-ITSL$_2$ transducer fragment that enforces Hungarian suffixal harmony

Interestingly, the desired 2+ threshold can be modelled using a multi-tiered transducer as the base. Suppose we have one tier V that tracks all vowels (i.e., including the transparent [i], [e], and [ɛ]), and another tier H that tracks only the harmony-triggering vowels (i.e., excluding the transparent [i], [e], and [ɛ]). This allows us to distinguish cases where the most recently produced vowel is a harmony trigger (i.e., the suffixes on V and H coincide) and cases where the most recently produced vowel is transparent but is itself preceded by a harmony trigger (i.e., the suffixes on V and H do not coincide). These two cases constitute two different states in the corresponding multi-tiered transducer, and by ensuring that forgetful transitions only originate from states where the two tier suffixes do not coincide, harmony will be obligatory across a single transparent vowel, although any additional transparent vowels will cause distance-

[5] We are assuming here for simplicity that harmony only affects underspecified suffix vowels, and that all base-internal vowels are fully specified in underlying forms.

based decay.

Consider now the transducer fragment in Table 1, which again determines the appropriate allomorph of the dative suffix /-nEk/ for bases containing only high vowels. Drawing the transducer in a legible manner is tricky, so we have opted to represent it as a collection of tables where each row corresponds to a transition. Rows are organized according to their origin state, and a label /a, b/ corresponds to having /a/ as the suffix on H (the tier of harmonic vowels) and /b/ as the suffix on V (the tier of all vowels). Notice how the /u,u/ and /u,i/ states both enforce harmony, but only the latter has a forgetful transition labelled 'i:i' (i.e. it has two rows for input /i/ in the table). Being in the /u,u/ state means that we have not read any transparent vowels after reading the most recent harmony-triggering vowel, while being in the /u,i/ state means that we have read *at least one* transparent vowel after reading the most recent harmony-triggering vowel. The transition labelled 'i:i' leaving /u,u/ and landing in /u,i/ does not have a forgetful counterpart, and so harmony remains obligatory across this one transparent vowel. The transition labelled 'i:i' leaving /u,i/ and looping back to /u,i/ does, however, have a forgetful counterpart. There is thus a chance that reading a second transparent vowel (and third, and fourth, etc.) will cause us to forget having read /u/. Forgetting that we read /u/ will bring us to the /λ,i/ state, which corresponds to thinking that we have not read a harmony trigger yet (or at the very least, not remembering the identity of the most recent harmony trigger). The decaying probability of harmony then results from the fact that it is increasingly unlikely to remain in the harmony-enforcing /u,i/ state as we read more and more transparent vowels.

Hayes and Londe (2006) and Hayes et al. (2009) collected a corpus of Hungarian noun bases (available at https://linguistics.ucla.edu/people/hayes/HungarianVH/index.htm), marking the percentage of times each base appears with the back versus front allomorph of the dative suffix (determined using Google search results). To ascertain whether and how much an MTSL model of decay outperforms a TSL model, we optimized a TSL-like transducer and an MTSL-like transducer against this corpus using a modified Baum-Welch algorithm. Unlike for the Malagasy tests above, we are not trying to maximize the probability of each datum in the sample, but trying to replicate the indicated probability of each datum as closely as possible. Each 'base + allomorph' combination with greater than 0 probability was thus treated as a separate datum, and the amount contributed by a reading step in that datum to a transition's estimated traversal count was multiplied by the datum's probability. Essentially, this would treat a training datum with an indicated probability of, for example, 0.75 as being 75% of a datum (i.e., a datum that was observed 0.75 times). Consequently, the optimization procedure is maximizing a weighted version of model log-likelihood rather than the regular log-likelihood. Where $P(o \mid i)$ is the probability of the input-output pair (i, o) as indicated in the sample S, and where $\hat{P}(o \mid i)$ is the probability of the input-output pair (i, o) predicted by the model M, regular and weighted log-likelihood can be expressed as in (3). There were only ever up to two output possibilities o_1 and o_2 for a given input string i,[6] and their observed probabilities $P(o_1 \mid i)$ and $P(o_2 \mid i)$ always summed to 1, as did their predicted probabilities $\hat{P}(o_1 \mid i)$ and $\hat{P}(o_2 \mid i)$. Maximizing the weighted log-likelihood ensures that we are on average minimizing the distance between the points $\langle \hat{P}(o_1 \mid i),\ \hat{P}(o_2 \mid i)\rangle$ and $\langle P(o_1 \mid i),\ P(o_2 \mid i)\rangle$ for the input strings in the sample.

(3) Regular model log-likelihood:

$$L(M \mid S) = \sum_{(i,o) \in S} \log(\hat{P}(o \mid i))$$

Weighted model log-likelihood:

$$L_W(M \mid S) = \sum_{(i,o) \in S} \log(\hat{P}(o \mid i)) \cdot P(o \mid i)$$

The TSL-like transducer had three states: /λ/ when there was no known preceding harmonic vowel, /F/ when the most recent harmonic vowel was front, and /B/ when the most recent harmonic vowel was back. It had forgetful transitions for /iː/, /i/, /eː/, and /ɛ/ leading to the /λ/ state out of the /B/ state. The /F/ state had no forgetful transitions since Hayes and Londe (2006) found that transparent vowels never block front harmonic vowels from imposing a front allomorph. The MTSL-like transducer had the 7 states listed in (4). The states /Biː/, /Bi/, /Beː/, and /Bɛ/ were kept separate, rather than having a single 'back + transparent' state since

[6] The one with the front allomorph of the dative suffix and the one with the back allomorph of the dative suffix.

Origin	Input	Output	Landing	Origin	Input	Output	Landing
/λ,λ/	/C/	[C]	/λ,λ/	/λ,i/	/C/	[C]	/λ,i/
/λ,λ/	/i/	[i]	/λ,i/	/λ,i/	/i/	[i]	/λ,i/
/λ,λ/	/y/	[y]	/y,y/	/λ,i/	/y/	[y]	/y,y/
/λ,λ/	/u/	[u]	/u,u/	/λ,i/	/u/	[u]	/u,u/
/λ,λ/	/⋉/	[nɛk]	NA	/λ,i/	/⋉/	[nɛk]	NA
Origin	Input	Output	Landing	Origin	Input	Output	Landing
/y,y/	/C/	[C]	/y,y/	/u,u/	/C/	[C]	/u,u/
/y,y/	/i/	[i]	/y,i/	/u,u/	/i/	[i]	/u,i/
/y,y/	/y/	[y]	/y,y/	/u,u/	/y/	[y]	/y,y/
/y,y/	/u/	[u]	/u,u/	/u,u/	/u/	[u]	/u,u/
/y,y/	/⋉/	[nɛk]	NA	/u,u/	/⋉/	[nɔk]	NA
Origin	Input	Output	Landing	Origin	Input	Output	Landing
/y,i/	/C/	[C]	/y,i/	/u,i/	/C/	[C]	/u,i/
/y,i/	/i/	[i]	/y,i/	/u,i/	/i/	[i]	/u,i/
/y,i/	/i/	[i]	/λ,i/	/u,i/	/i/	[i]	/λ,i/
/y,i/	/y/	[y]	/y,y/	/u,i/	/y/	[y]	/y,y/
/y,i/	/u/	[u]	/u,u/	/u,i/	/u/	[u]	/u,u/
/y,i/	/⋉/	[nɛk]	NA	/u,i/	/⋉/	[nɔk]	NA

Table 1: A quasi-IMTSL$_2$ transducer fragment that enforces Hungarian suffixal harmony

Hayes and Londe (2006) found that the height of the most recent transparent vowel has a significant effect on the probability of a back allomorph. Each of the states /Biː/, /Bi/, /Be/, and /Bɛ/ had forgetful transitions for /iː/, /i/, /e/, and /ɛ/ leading to the state /N/.

(4) States in the MTSL-like Hungarian transducer

- /λ/ = no preceding vowel OR the most recent vowel is transparent with no known preceding harmonic vowel
- /F/ = the most recent harmonic vowel is front
- /BB/ = the most recent vowel is back
- /Biː/ = the most recent harmonic vowel is back but the most recent vowel is iː
- /Bi/ = the most recent harmonic vowel is back but the most recent vowel is i
- /Beː/ = the most recent harmonic vowel is back but the most recent vowel is e
- /Bɛ/ = the most recent harmonic vowel is back but the most recent vowel is ɛ

Unfortunately, a log-likelihood ratio test cannot compare the two models because their parameters are not strictly nested; none of their forgetful transitions originate from equivalent states. Accordingly, their relative performance was assessed using their Akaike Information Criterion (AIC). Lower AIC values are preferred, and a model's AIC is equal to 2 times its number of free parameters minus 2 times its log-likelihood. The TSL-like model had a weighted log-likelihood of -266.90 and had 7 free parameters, so its AIC is 547.8. For its part, the MTSL-like model had a weighted log-likelihood of -249.73 and had 23 free parameters, so its AIC is 545.46. Going from the TSL model to the MTSL model reduces AIC by 2.34, which is not substantial, but nonetheless favours the MTSL model.

A reviewer points out that the closely related Bayesian Information Criterion (BIC) will likely heavily favor the TSL model as opposed to the MTSL model, even though both criteria tend to favour the same models in practice. The BIC more harshly penalizes parameter count: it is obtained by multiplying number of free parameters by the natural logarithm of the sample size and then subtracting 2 times the log-likelihood. There were 9427 input-output pairs in the training sample, so the TSL model has a BIC of 597.86 and the MTSL model has a BIC of 709.94; the reviewer thus correctly speculates that BIC vastly prefers the TSL model over the MTSL one. We are unsure of how to reconcile the discrepancy between the opposite preferences of AIC and BIC in this case, but we lean towards siding with the BIC's preference since it is much stronger. Nevertheless, it is worth examining where the MTSL model's higher accuracy comes from, since it may be worthwhile in other cases.

Visual inspection of the weights in the optimized transducers suggests that the greater accuracy of the MTSL model comes from the fact that it can distinguish between spans of 1 versus $2+$ transparent vowels, a distinction not possible in the TSL model. For example, the MTSL model assigns a probability of about $0.97 * 0.55 = 0.53$ to the form [ɔspirin-nɔk] 'aspirin-DAT' since the probability of harmony while in the state /Bi/ (i.e., across a single instance of /i/) is about 0.97 and the probability of /i/ not causing forgetfulness out of the state /Bi/ is about 0.55. Compare this to the TSL transducer which assigns the same form a probability of about $0.99 * 0.93^2 = 0.86$ since the probability of harmony while in the state /B/ is about 0.99 and the probability of /i/ not causing forgetfulness out of the state /B/ is about 0.93. The actually observed frequency of harmony for this noun is 0.21 and so the MTSL transducer, while still a fair ways off, is much closer than the TSL transducer.

Consistent with this interpretation of the models' differing performance, Table 2 compares the observed average probability of harmony against the average probabilities predicted by the TSL and MTSL models, broken down by the number of intervening transparent syllables. Taking every noun for which back a back allomorph is possible (i.e., whose rightmost harmonic vowel is back), we find that adjacent harmony has an average probability of 0.99 (5317 eligible nouns), harmony across a single transparent vowel has an average probability of 0.67 (370 eligible nouns), harmony across two transparent vowels has an average probability of 0.18 (63 eligible nouns) and harmony across three transparent vowels has an average probability of 0.00 (8 eligible nouns). In particular, we see that the TSL model overestimates the probability of harmony across two transparent vowels, whereas the MTSL model closely matches the observed probabilities in all cases.

Interestingly, both the trained TSL model and the trained MTSL model reproduce an additional aspect of Hungarian harmony whereby vowel height gradiently affects the degree to which a front unrounded vowel is transparent. Specifically, lower front unrounded vowels are "less transparent" than higher front unrounded vowels (Hayes and Londe, 2006; Hayes et al., 2009; Kimper, 2011; Rebrus and Törkenczy, 2016; Ozburn, 2019). In the optimized TSL model, the forgetfulness parameters out of state /B/ for /i/, /iː/, /eː/, and /ɛ/ are weighted 0.065, 0.11, 0.23, and 0.91 respectively, so lower vowels cause more forgetfulness than higher vowels. In the optimized MTSL model, the probability of appending the back allomorph upon ending in the /Bi/, /Biː/, /Beː/ and /Bɛ/ states is respectively 0.97, 0.99, 0.80, and 0.11, so backness harmony is more likely across a higher vowel than a lower vowel. The same height effect is also somewhat apparent in the MTSL model's forgetfulness parameters: the parameters for /iː/, /eː/, and /ɛ/ have average weights of 0.51, 0.78, and 0.91 respectively, mimicking the trend in the TSL model's forgetfulness parameters. The mimicking is not perfect, however, as the average forgetful weight for the vowel /i/ is unexpectedly high at 0.8. This mismatch could perhaps be because back vowels are only uncommonly followed by a chain of multiple front unrounded vowels, making the weights of the MTSL model's forgetfulness parameters a less reliable reflection of the height effect. Indeed, there were cases where the forgetful and non-forgetful transitions for the same vowel out of the same state both had a weight of 0, meaning that neither transition was ever crossed by the sample. These transitions effectively do not exist and were not considered when calculating the average weights.

Transparent syllables	Observed average	TSL average	MTSL average
0	0.99	0.99	0.99
1	0.67	0.64	0.67
2	0.18	0.33	0.19
3	0.00	0.03	0.01

Table 2: Average probability of Hungarian harmony by number of transparent syllables

5 Conclusion

In the existing computational work that models long-distance phonological processes using tiers, there is the tacit and convenient assumption that the processes are an all or nothing affair, but long-distance processes are often optional in reality. We showed here that probabilistic tier-based transducers with a structure similar to that of their categorical counterparts can capture gradient application while maintaining the relative computational simplicity afforded to us by tier-based strict locality.

In particular, we demonstrated that distance-based decay can be modeled by strategically adding duplicate non-tier transitions that make the transducer forget the identity of the most recent tier-element, in a sense causing the machine's memory to deteriorate over time. Using established techniques for optimizing the weights of probabilistic automata, we found that these models performed well relative to two real-language data sets. The Malagasy case study suggested that the presence and strength of forgetful transitions might be tied to similarity. For its part, he Hungarian case study showed that an MTSL model can distinguish between spans of 1 versus 2+ transparent elements, which may be a useful ability for some patterns. Finally, it should be said that all of the simulations presented above assume that the necessary tiers are known in advance, although phonological learning ideally involves as little *a priori* knowledge as possible. Methods exist for learning the tier of a TSL_2 function efficiently from positive data (Burness and McMullin, 2019), although it remains to be seen whether they can be adapted to probabilistic cases.

References

Samuel Andersson, Hossep Dolatian, and Yiding Hao. 2020. Computing vowel harmony: The generative capacity of search and copy. In *Proceedings of the 2019 Annual Meeting on Phonology*.

Leonard E. Baum. 1972. An inequality and associated maximization technique occurring in the statistical estimation for probabilistic functions of Markov processes. *Inequalities*, 3:1–8.

Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41:164–171.

Phillip Burness and Kevin McMullin. 2019. Efficient learning of Output Tier-Based Strictly 2-Local functions. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 78–90. Association for Computational Linguistics.

Phillip Burness and Kevin McMullin. 2020. Multi-tiered strictly local functions. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 245–255. Association for Computational Linguistics.

Jane Chandlee. 2014. *Strictly Local Phonological Processes*. Doctoral dissertation, University of Delaware.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2014. Learning Strictly Local subsequential functions. *Transactions of the Association for Computational Linguistics*, 2:491–503.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2015. Output Strictly Local functions. In *Proceedings of the 14th Meeting on the Mathematics of Language (MOL 2015)*, pages 112–125.

Jane Chandlee and Jeffrey Heinz. 2018. Strict Locality and phonological maps. *Linguistic Inquiry*, 49:23–60.

Jane Chandlee, Jeffrey Heinz, and Adam Jardine. 2018. Input strictly local opaque maps. *Phonology*, 35:171–205.

Andries Coetzee and Joe Pater. 2011. The place of variation in phonological theory. In John Goldsmith, Jason Riggle, and Jason Yu, editors, *The Handbook of Phonological Theory*, 2nd edition, pages 401–431. Blackwell.

André Coupez. 1980. *Abrège de Grammaire Rwanda*. Institut National de Recherche Scientifique, Butare.

Kornelis F. de Blois. 1975. *Bukusu Generative Phonology an Aspects of Bantu Structure*. Number 85 in Annales. Musée Royal de l'Afrique Centrale, Tervuren.

Jean-Marie de la Beaujardière. 2004. Malagasy dictionary and encyclopedia of Madagascar.

Colin de la Higuera. 2010. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press, New York.

John Goldsmith. 1976. *Autosegmental Phonology*. Ph.D. thesis, MIT.

Sharon Goldwater and Mark Johnson. 2003. Learning OT constraint rankings using a maximum entropy model. In *Proceedings of the Workshop on Variation within Optimality Theory*, pages 111–120.

Gunnar Ólafur Hansson. 2010. *Consonant Harmony: Long-Distance Interaction in Phonology*. Number 145 in University of California Publications in Linguistics. University of California Press, Berkeley, CA.

Yiding Hao and Samuel Andersson. 2019. Unbounded stress in subregular phonology. In *Proceedings of the 16th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology and Morphology*, pages 135–143, Florence, Italy. Association for Computational Linguistics.

Yiding Hao and Dustin Bowers. 2019. Action-sensitive phonological dependencies. In *Proceedings of the 16th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology and Morphology*, pages 218–228, Florence, Italy. Association for Computational Linguistics.

Bruce Hayes and Zsuzsa Londe. 2006. Stochastic phonological knowledge: The case of Hungarian vowel harmony. *Phonology*, 23:59–104.

Bruce Hayes and Colin Wilson. 2008. A maximum entropy model of phonotactics and phonological learning. *Linguistic Inquiry*, 39:379–440.

Bruce Hayes, Kie Zuraw, Peter Siptar, and Zsuzsa Londe. 2009. Natural and unnatural constraints in Hungarian vowel harmony. *Language*, 85:822–863.

Peter Jurgec. 2011. *Feature Spreading 2.0: A Unified Theory of Assimilation*. Doctoral dissertation, University of Tromso.

Alexandre Kimenyi. 1979. *Studies in Kinyarwanda and Bantu Phonology*. Linguistic Research, Carbondale, IL.

Wendell A. Kimper. 2011. *Competing Triggers: Transparency and Opacity in Vowel Harmony*. Doctoral dissertation, University of Massachusetts Amherst.

Andrew Martin. 2005. *The Effects of Distance on Lexical Bias: Sibilant Harmony in Navajo Compounds*. Master's thesis, University of California, Los Angeles.

David Odden. 1994. Adjacency parameters in phonology. *Language*, 70:289–330.

Avery Ozburn. 2019. A target-oriented approach to neutrality in vowel harmony: Evidence from Hungarian. *Glossa*, 4:1–36.

Péter Rebrus and Miklós Törkenczy. 2016. A non-cumulative pattern in vowel harmony: A frequency-based account. In *Proceedings of the 2015 Annual Meeting on Phonology*.

Enrique Vidal, Franck Tollard, Colin de la Higuera, Francisco Casacuberta, and Rafael Carrasco. 2005a. Probabilistic finite-state machines - Part I. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1013–1025.

Enrique Vidal, Franck Tollard, Colin de la Higuera, Francisco Casacuberta, and Rafael Carrasco. 2005b. Probabilistic finite-state machines - Part II. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1026–1039.

Rachel Walker, Dani Byrd, and Fidèle Mpiranya. 2008. An articulatory view of Kinyarwanda coronal harmony. *Phonology*, 25:499–535.

Rachel Walker and Fidèle Mpiranya. 2006. On triggers and opacity in coronal harmony. In *Proceedings of the 31stth Annual Meeting of the Berkeley Linguistics Society*, University of California, Berkeley.

Jesse Zymet. 2015. Distance-based decay in long-distance phonological processes. In *Proceedings of the 32nd West Coast Conference on Formal Linguistics*, pages 72–81, Sommerville, MA. Cascadilla Press.

Jesse Zymet. 2020. Malagasy ocp targets a single affix: Implications for morphosyntactic generalization in learning. *Linguistic Inquiry*, 51:624–634.