

Instructions for ACL 2020 Proceedings

Anonymous ACL submission

Abstract

1 Introduction

Formal language theory has long been used to study the complexity of linguistic dependencies. Recent research in this sense has posited that the phonotactics of natural languages can be described by subclasses of the regular languages. In particular, tier-based strictly local (TSL) grammars — a minor extension of n -gram models — have been shown to be able to capture a variety of non-local, unbounded processes (Heinz et al., 2011; McMullin, 2016; McMullin and Hansson, 2016). Recently however, it has been suggested that the particular notion of relativized locality employed by the TSL class is unable to describe a variety of complex phonotactic patterns cross-linguistically. Based on this linguistic motivation, extensions have been proposed in the search of the right fit for natural language phonotactics. Specifically, input-sensitive TSL languages have been suggested as being able to encode a combination of local and non local requirements on the well-formedness of strings in the language.

Apart from typological coverage, an important aspect of evaluating the linguistic relevance of these analyses is to understand under which conditions such patterns are learnable. In this sense, an approach to learning grounded in grammatical inferences is interesting, as it illuminates how properties of the patterns can restrict the learning space in useful ways. In this framework, TSL languages have been shown to be efficiently learnable from positive input only. While ITSL languages have been argued to share the same property, no learning algorithm exists for this class. In this paper, we extend McMullin et al. (2019) inference algorithm for multiple tier-based strictly 2 local languages

(MITSL₂), in order to learn patterns in the intersection closure of ITSL₂ which consider 2-local contexts for segments in the input string (MITSL₂²). The intersection closure is essential, if we strive to provide learning approaches able to capture the whole phonotactics of a language, and not one single pattern at the time. We evaluate our algorithm qualitatively over a variety of natural and formal examples, and discuss known limitations of the framework and possible extensions.

2 MITSL Languages and Linguistic Motivation

Many dependencies in phonology can be captured by SL grammars: *local constraints* that only make distinctions on the basis of contiguous substrings of segments up to some length k (essentially, k -grams; Heinz, 2011). For example, a ($k=2$) local dependency requiring /s/ to surface as [z] when followed by [l] can be captured by a grammar that forbids the sequence [sl]. However, (unbounded) long-distance dependencies cannot be captured by local constraints, and have been characterized instead as *tier-based strictly local*.

Tier-based strictly local languages (TSL) are able to encode a notion of relativized locality inspired by the idea of phonological tier, already popular in autosegmental phonology (Goldsmith, 1976). While a formal introduction to the properties of TSL is beyond the scope of this paper, a TSL dependency is intuitively non-local in the input string but local over a *tier*. A tier is defined as the projection of a subset of the segments of the input string, and the grammar constraints are characterized as the set of sequences of length k not allowed on the tier. For instance, the example in Figure 1 (from Aari, an Omotic language of south Ethiopia) shows how to enforce long-distance sibilant harmony in anteriority. First one projects from the string a tier T

that only contains sibilants, and then one bans contiguous [ʒs] and [sʒ] on T (see (Hayward, 1990)).

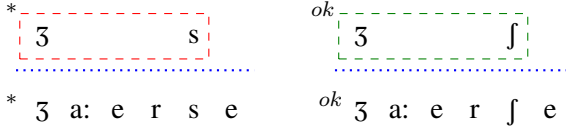


Figure 1: Example of sibilant harmony over tier from Aari.

The class of TSL languages has been shown to have good cross-linguistic coverage, accounting for a variety of different phonotactic patterns cross-linguistically (Heinz et al., 2011; McMullin, 2016; Graf, 2017). Moreover, and most interesting to us, TSL_k languages have been shown to be efficiently (polynomial in time and input) learnable in the limit from positive data, even when the tier-alphabet is not known *a priori* (Jardine and Heinz, 2016; Jardine and McMullin, 2017).

However, there are two main known limits to TSL as a good formal account for natural language phonotactics.

First, it is known that TSL languages are not closed under intersection. Lack of closure under intersection is problematic as it entails that the complexity of phonological dependencies is no longer constant under factorization. This implies that the upper bound for phonological phenomena would shift, depending on whether one treats a constraint as a single phenomenon or the interaction of multiple phenomena. Moreover, we clearly want to be able to consider multiple phenomena at the same time when describing the phonotactics of a language. Intersection closure is thus a fundamentally desirable property from a linguistic perspective. To account for this, De Santo and Graf (2019) propose the multiple tier-based strictly local (MTSL) class, as a proper extension of TSL formalizing its intersection closure. Intuitively, MTSL can be conceptualized as a class encoding multiple projections (tiers) at the same time, and enforcing distinct strictly local constraints over each tier. McMullin et al. (2019) propose an algorithm that efficiently learns multiple tier-based strictly 2-local (i.e. where tier constraints are bigrams) dependencies, with no a-priori knowledge about the tier-segments or the number of tiers required.

The second limit of TSL lies in the simplicity of its projection mechanism. Recently, several patterns have been reported that cannot be described by the way TSL currently uses tier projection to mask out parts of a string before enforcing some strictly local constraint (McMullin, 2016; Mayer and Major, 2018; Baek, 2017; Graf and Mayer, 2018; De Santo and Graf, 2019). These patterns include the long-distance sibilant harmony in Imdlawn Tashlhiyt (McMullin, 2016), the nasal harmony pattern in Yaka (Walker, 2000), the unbounded stress of Classical Arabic (see (Baek, 2017) and references therein), and cases of unbounded tone plateauing. These patterns share the common trait that one has to inspect the local context of a segment before projecting it on a tier.

Consider the case of Consonantal Nasal harmony in Yaka, in which a nasal stop induces nasalization of voiced consonants occurring at any distance to its right (??). For instance, the segmental alternation shown in Ex. (1) is due to the phoneme /d/ surfacing as [n] after a preceding nasal (cf. Ex. (1a, 1b vs. 1c)). Vowels and voiceless consonants intervening between the two harmonizing stops remain unaffected (cf. Ex. (2)).

- (1) a. yán-ini ‘to cry out’
b. yád-idi ‘to spread’
c. *yán-idi
- (2) a. hámúk-ini ‘to give away’
b. míituk-ini ‘to sulk’
- (3) a. bíimb-idi ‘to embrace’
b. kúúnd-idi ‘to bury’
c. nááŋg-ini ‘to last’

A TSL analysis for this pattern seems straightforward, as this data can be captured by projecting a tier of voiced consonants, and enforcing constraints banning tier adjacent [nd].

However, observe now the examples in Ex. (3): consonantal complexes composed of a nasal and a voiced oral stop neither trigger Ex. (3a,3b) nor block nasality agreement Ex. (3c). Fig. 2 exemplifies why this interaction of a local and a non-local dependency is not TSL. Since [nd] is sometimes observed in a string-adjacent context (as in Ex. (3b)), it must be permitted as a 2-gram on a tier — even though it is only allowed when [nd] re immediately adjacent in the string. But then, a TSL

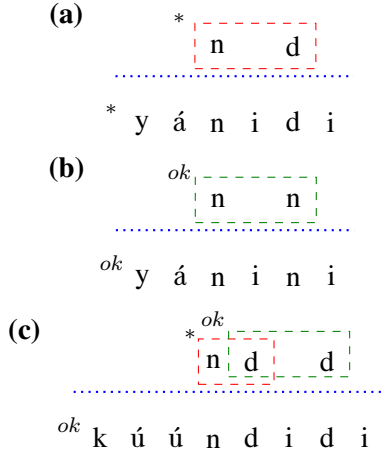


Figure 2: Example of a TSL analysis of nasal harmony in Yaka: (a) is ill-formed because of tier adjacent *[nd]; (b) is well-formed since there are no voiced stops on the tier disagreeing in nasality; (c) is well-formed because the [d] immediately following [n] stops the latter from being a trigger for harmony, but it is still ruled out by the constraint needed for (b).

grammar would have no means of distinguishing Ex. (1b) from Ex. (3b).

The reader might point out that the difference between Fig. 2.a and Fig. 2.c can be resolved by extending the tier-grammar to consider 3-grams. However, in order to enforce harmony correctly, the tier-projection places every occurrence of voiced stops in the string on the tier, thus making 3-grams constraints insufficient (e.g., Ex. (3c)). Moreover, since the number of segments between harmonizing elements is potentially unbounded, no TSL grammar can generally account for this pattern, independently of the dimension of the tier k -grams.

Let us consider the examples in Ex. (3) once more. Any nasal immediately followed by a voiced stop does not trigger harmony. In fact, since they do not block the harmonic process, neither the nasal nor the stop participate in the harmony at all. If we could make the projection of nasals and stops avoid those segments that appear in specific consonant clusters (e.g. [nd]) the tier constraints discussed above would work once again. This is not possible with TSL as originally defined in (Heinz et al., 2011), as TSL selects tier elements only based on their 1-local properties (i.e. which kind of segment they are). However, this kind of expressivity can be accomplished by increasing the locality window of the *tier projection mechanism*.

In particular, the intuition behind De Santo and

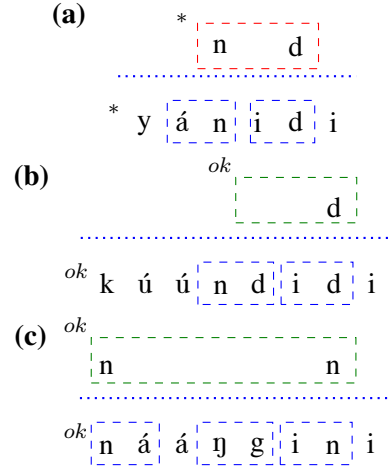


Figure 3: Example of a ITSL analysis of nasal harmony in Yaka: (a) is ill-formed because of adjacent *[nd]; (b) is well-formed since [n] is followed by another [n] later in the string; (c) is well-formed because the [nd] cluster does not enforce nasality on the following stops. Note that [n,d] are projected on the tier only when not immediately adjacent in the input.

Graf (2019)’s ITSL class is that a TSL grammar can be made simultaneously aware of local and non-local properties of segments in the string with a natural change to the definition of the erasing function.

Fig. 3 shows how, by increasing the locality of the projection to 2, we allow the grammar to project a nasal iff it is not immediately followed by a voiced oral stop, and then use 2-local tier constraints to ban nd. This time, possible intermediate clusters are not a problem, since the projection is able to infer that they are in local contexts that make them irrelevant to the harmonic process.

Finally, note the following additional data:

- (4) a. kém-ene
- b. kéb-edé

Ex. (4) shows a vowel alternation that is independent of the nasality process, and is instead due to vowel height harmony. Vowel harmony can be easily accounted for with a TSL grammar. However, this is only true if we analyze it by itself, and fails if we try to model nasal harmony and vowel harmony in a single grammar. In order to account for this, vowel will need to be projected on the tier, thus interfering with the nasalization process. This issue is resolved working with the intersection closure of ITSL languages (MITSL).

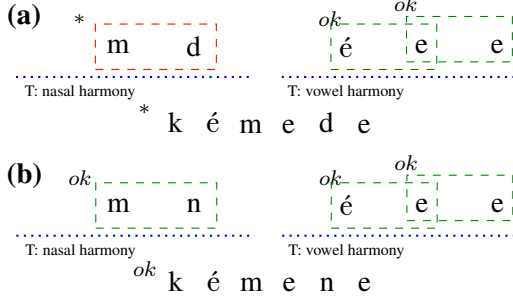


Figure 4: Example of a MITSL analysis of Yaka nasal and vowel harmony: (a) is ill-formed because there is a violation on the nasal harmony tier; (b) is well-formed since there are no violations on either tier.

This class can be intuitively understood as having a grammar projecting multiple tiers, and enforcing independent local constraints on each tier. For a string to belong to the language, it needs to be well-formed on every tier. For instance, Fig. 4 shows a grammar projecting two separate tier: a tier of vowel, with constraints ensuring height harmony: and a tier enforcing nasal harmony as in the examples above.

In the rest of the paper, we directly expand of McMullin et al. (2019)’s MTSL2IA algorithm and present a grammatical inference algorithm able to learn conjunction ITSL grammars with 2-local contexts and 2-local tier constraints (k -MITSL₂), only from positive examples and without a-priori knowledge about the content or the number of tiers.

3 MITSL Inference Algorithm

The remainder of the paper discusses our learning algorithm for MITSL languages with projection contexts and tier constraints of size 2 (MITSL₂). While the previous section presented an intuitive definition of MITSL languages, a more formal definition is necessary in order to understand the way the algorithm works. Thus, we first introduce some mathematical preliminaries and discuss how the definition of MITSL grammar presented in (De Santo and Graf, 2019) grounds the intuition behind our generalization of McMullin et al. (2019)’s learning algorithm. We also discuss a generalization of the notion of 2-path as introduced by ?.

3.1 Formal Preliminaries

We assume familiarity with set notation on the reader’s part. Given a finite alphabet Σ , Σ^* is the set of all possible finite strings of symbols drawn

from Σ . A language L is a subset of Σ^* . For every string w and every non-empty string u , $|w|$ denotes the length of the string, $|w|_u$ denotes the number of occurrences of u in w , and λ is the unique empty string. Left and right word boundaries are marked by \bowtie , $\bowtie \notin \Sigma$ respectively.

A string u is a k -factor of a string w iff $\exists x, y \in \Sigma^*$ such that $w = xuy$ and $|u| = k$. The function fac_k maps words to the set of k -factors within them: $\text{fac}_k(w) := \{u : u \text{ is a } k\text{-factor of } w \text{ if } |w| \geq k, \text{ else } u = w\}$. For example, $\text{fac}_2(aab) = \{aa, ab\}$. The domain of fac_k is generalized to languages $L \subseteq \Sigma^*$ in the usual way: $\text{fac}_k(L) = \bigcup_{w \in L} \text{fac}_k(w)$.

As usual, we allow standard Boolean connectives ($\wedge, \vee, \neg, \rightarrow$), and first-order quantification (\exists, \forall) over individuals. We let $x \prec y$ denote *precedence*, $x \approx y$ denote *identity*, and x, y denote variables ranging over positions in a finite string $w \in \Sigma^*$. Note that \prec is a strict total order. The remaining logical connectives are obtained from the given ones in the standard fashion, and brackets may be dropped where convenient. For example, *immediate precedence* is defined as $x \triangleleft y \leftrightarrow x \prec y \wedge \neg \exists z [x \prec z \wedge z \prec y]$.

TSL languages were defined by as a class where k -local constraints only apply to elements of a tier $T \subseteq \Sigma$. In order to do so, an erasing function (also called projection function hereafter) is introduced to delete (or mask) all symbols that are not in T . In order to extend the notion of tier in TSL languages to consider local properties of the segments in the input string, De Santo and Graf (2019) take inspiration from (Chandlee and Heinz, 2018) and define ITSL projection function in terms of local contexts.

Definition 1 (Contexts). A k -context c over alphabet Σ is a triple $\langle \sigma, u, v \rangle$ such that $\sigma \in \Sigma$, $u, v \in \Sigma^*$ and $|u| + |v| \leq k$. A k -context set is a finite set of k -contexts.

Definition 2 (ISL Projection). Let C be a k -context set over Σ (where Σ is an arbitrary alphabet also containing edge-markers). Then the input strictly k -local (ISL- k) tier projection π_C maps every $s \in \Sigma^*$ to $\pi'_C(\bowtie^{k-1}, s\bowtie^{k-1})$, where $\pi'_C(u, \sigma v)$ is defined as follows, given $\sigma \in \Sigma \cup \{\varepsilon\}$ and $u, v \in \Sigma^*$:

$$\begin{aligned} \varepsilon & \quad \text{if } \sigma av = \varepsilon, \\ \sigma \pi'_C(u\sigma, v) & \quad \text{if } \langle \sigma, u, v \rangle \in C, \\ \pi'_C(u\sigma, v) & \quad \text{otherwise.} \end{aligned}$$

Note that an ISL-1 tier projection only determines projection of σ based on σ itself, showing that this projection function is really just an extension of what happens for TSL languages. The definition of ITSL languages then is as follows.

Definition 3 (ITSL). A language L is m -input local k -TSL (m -ITSL $_k$) iff there exists an m -context set C and a finite set $S \subseteq \Sigma^k$ such that

$$L = \{w \in \Sigma^* : F_k(\bowtie^{k-1} \pi_C(w) \bowtie^{k-1}) \cap S = \emptyset\}.$$

A language is input-local TSL (ITSL) iff it is m -ITSL $_k$ for some $k, m \geq 0$. We call $\langle S, C \rangle$ an ITSL grammar.

Note that the notion of tier is here expressed by the set of contexts C , which is the set of tier segments with the locality conditions necessary for them to be relevant to the tier constraints.

Let us return to the interaction of local dissimilation and non-local harmony in Samala. This process can be handled by an 2-ITSL $_3$ grammar $\langle S, C \rangle$ with

- $S := \{sf, fs, snx\}$ where $x \in \{\Sigma - s\}$,
- C contains all of the following contexts, and only those:
 - $\langle s, \varepsilon, \varepsilon \rangle$
 - $\langle S, \varepsilon, \varepsilon \rangle$
 - $\langle n, s, \varepsilon \rangle$

To do so, the algorithm incorporates the notion of a 2-path (?).

Intuitively, a 2-path can be thought of as a precedence relation $(\rho_1 \dots \rho_2)$ accompanied by the set X of symbols that intervene between ρ_1 and ρ_2 . Formally, each 2-path is therefore a 3-tuple of the form $\langle \rho_1, X, \rho_2 \rangle$. For example, the string *abcc* includes the following 2-paths: $\langle a, \emptyset, b \rangle, \langle a, \{b\}, c \rangle, \langle a, \{b, c\}, c \rangle, \langle b, \emptyset, c \rangle, \langle b, \{c\}, c \rangle$.

3.2 The Algorithm

The algorithm exploits the fact that if a bigram $\rho_1 \rho_2$ is banned on some tier, then it will never appear in string-adjacent contexts. For each $\rho_1 \rho_2$ absent from the training data, the goal is therefore to determine which segments can be safely removed from the associated tier. In short, by examining the set of 2-paths present in the training data allows, we can determine which segments are freely distributed with respect to a bigram $\rho_1 \rho_2$ that is

Data: A finite input sample $I \subset \Sigma^*$

Result: MITSL $_2^2$ grammar of the form

$$G = \bigwedge \langle T_i, R_i \rangle$$

Initialize $F = \text{fac}_4(\Sigma^*) - \text{fac}_4(I)$;

Initialize $B = \text{fac}_2(\Sigma^*)$;

foreach $f \in F$ **do**

Initialize $R_i = f, T_i = B$; (with

$1 \leq i \leq |F|$)

foreach $\sigma \in B - \{f[:2], f[2:]\}$ **do**

if $\forall \langle f[:2], X, f[2:] \rangle \in \text{paths}_2(I)$ s.t.

$\sigma \in X, \langle f[:2], X - \{\sigma\}, f[2:] \rangle \in$

$\text{paths}_2(I)$

then $T_i = T_i - \{\sigma\}$ (i.e., remove σ from T_i);

end

$G_i = \langle T_i, R_i \rangle$

end

Return $G = G_1 \wedge G_2 \wedge \dots \wedge G_{|F|}$

Algorithm 1: Pseudocode for the MITSL $_2^2$ Inference Algorithm introduced in this paper.

known to be banned on some tier. Specifically, if all of the attested $\langle \rho_1, X, \rho_2 \rangle$ 2-paths that include an intervening σ are likewise attested *without* an intervening σ , the algorithm removes σ from the tier, since the presence of $\rho_1 \dots \rho_2$ is not dependent on an intervening σ .

3.3 Qualitative Evaluation: An Artificial MITSL $_2^2$ Pattern

3.4 Unlearnable Patterns

However, we note that the algorithm relies on the assumption that each bigram restriction is enforced on at most one tier. A small portion of logically-possible MTSL patterns therefore remains out of reach at present, but the problematic cases are among those which ? claim to be unattested (those with overlapping tiers, such that $T_1 \not\subseteq T_2$ and $T_1 \cap T_2 \neq \emptyset$). Specifically, the MTSL2IA fails if these overlapping tiers are associated with a single $\rho_1 \rho_2$ restriction (i.e., when it is blocked by a different symbol on each tier), but it will succeed when they are associated with different restrictions.

4 Extending the Evaluation

5 Conclusion

References

Hyunah Baek. 2017. Computational representation of unbounded stress patterns: tiers with structural features.

In *Proceedings of the 53rd Meeting of the Chicago Linguistic Society (CLS53)*.

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

Aniello De Santo and Thomas Graf. 2019. Structure sensitive tier projection: Applications and formal properties. In *International Conference on Formal Grammar*, pages 35–50. Springer.

John Goldsmith. 1976. *Autosegmental phonology*. Ph.D. thesis, MIT, Cambridge, MA.

Thomas Graf. 2017. [The power of locality domains in phonology](#). *Phonology*, 34:385–405.

Thomas Graf and Connor Mayer. 2018. Sanskrit n-retroflexion is input-output tier-based strictly local. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 151–160.

R. J. Hayward. 1990. Notes on the Aari language. In R. J. Hayward, editor, *Omotic language studies*, pages 425–493. School of Oriental and African Studies, U. of London, London, UK.

Jeffrey Heinz. 2011. Computational phonology – part 1: Foundations. *Language and Linguistics Compass*, 5(4):140–152.

Jeffrey Heinz, Chetan Rawal, and Herbert Tanner. 2011. [Tier-based strictly local constraints for phonology](#). In *Proceedings of the ACL 49th: Human Language Technologies: Short Papers - Volume 2*, pages 58–64.

Adam Jardine and Jeffrey Heinz. 2016. [Learning tier-based strictly 2-local languages](#). *Transactions of the ACL*, 4:87–98.

Adam Jardine and Kevin McMullin. 2017. Efficient learning of tier-based strictly k -local languages. In *Language and Automata Theory and Applications, 11th International Conference, LNCS*, pages 64–76. Springer.

Connor Mayer and Travis Major. 2018. A challenge for tier-based strict locality from Uyghur backness harmony. In *Formal Grammar 2018. Lecture Notes in Computer Science*, vol. 10950, pages 62–83. Springer, Berlin, Heidelberg.

Kevin McMullin. 2016. [Tier-based locality in long-distance phonotactics?: learnability and typology](#). Ph.D. thesis, U. of British Columbia.

Kevin McMullin, Alëna Aksënova, and Aniello De Santo. 2019. [Learning phonotactic restrictions on multiple tiers](#). *Proceedings of the Society for Computation in Linguistics*, 2(1):377–378.

Kevin McMullin and Gunnar Hansson. 2016. [Long-distance phonotactics as tier-based strictly 2-local languages](#). *Proceedings of the Annual Meetings on Phonology*, 2(0).

Rachel Walker. 2000. [Yaka nasal harmony: Spreading or segmental correspondence?](#) *Annual Meeting of the Berkeley Linguistics Society*, 26(1):321–332.

A Supplemental Material