Online Learning of ITSL Grammars

Jacob K. Johnson & Aniello De Santo

University of Utah
aniello.desanto@utah.edu
https://aniellodesanto.github.io/

SCiL 2024

Online Learning of ITSL Grammars

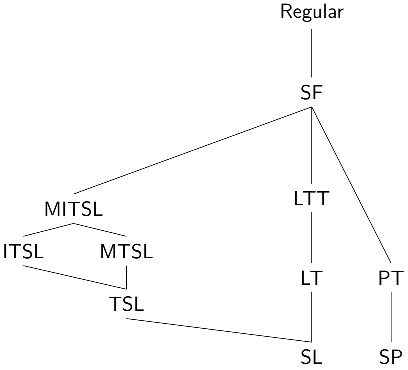
Jacob K. Johnson & Aniello De Santo

University of Utah
aniello.desanto@utah.edu
https://aniellodesanto.github.io/

SCiL 2024

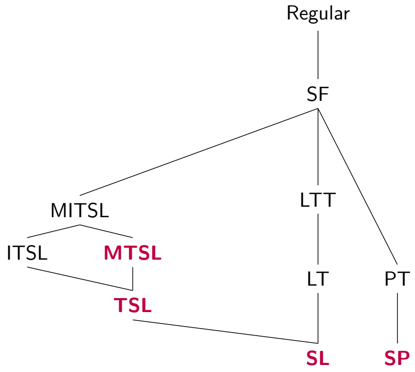
The Talk in One Slide

Phonotactics as a **subregular system**.
(McNaughton & Papert 1971, Rogers et al 2010, Heinz 2015, McMullin 2016, De Santo & Graf 2019 a.o.)



The Talk in One Slide

Phonotactics as a **subregular system**.
(McNaughton & Papert 1971, Rogers et al 2010, Heinz 2015, McMullin 2016, De Santo & Graf 2019 a.o.)

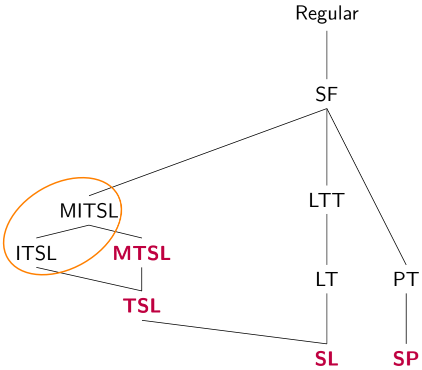


- Learnable? Jardine & Heinz (2016), Jardine & McMullin (2017), McMullin et al. (2019), Lambert (2021) a.o.

The Talk in One Slide

Phonotactics as a **subregular system**.

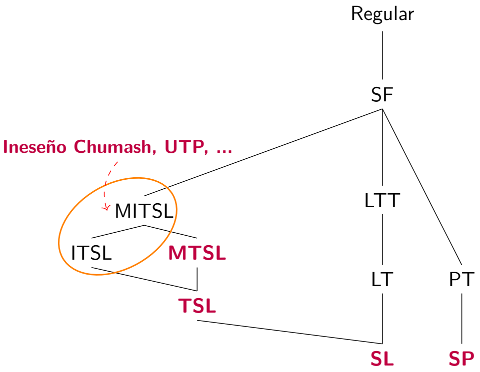
(McNaughton & Papert 1971, Rogers et al 2010, Heinz 2015, McMullin 2016, De Santo & Graf 2019 a.o.)



- Learnable? Jardine & Heinz (2016), Jardine & McMullin (2017), McMullin et al. (2019), Lambert (2021) a.o.

The Talk in One Slide

Phonotactics as a **subregular system**.
(McNaughton & Papert 1971, Rogers et al 2010, Heinz 2015, McMullin 2016, De Santo & Graf 2019 a.o.)



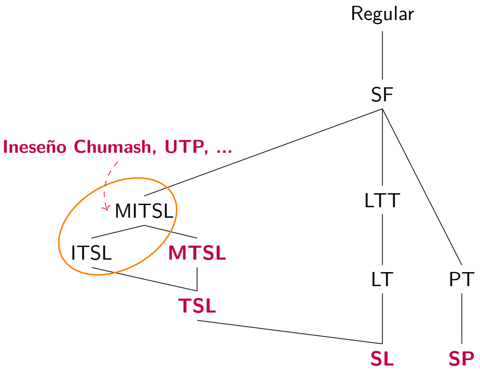
- **Learnable?** Jardine & Heinz (2016), Jardine & McMullin (2017), McMullin et al. (2019), Lambert (2021) a.o.

The Talk in One Slide

Phonotactics as a **subregular system**.
(McNaughton & Papert 1971, Rogers et al 2010, Heinz 2015, McMullin 2016, De Santo & Graf 2019 a.o.)

Goal today!

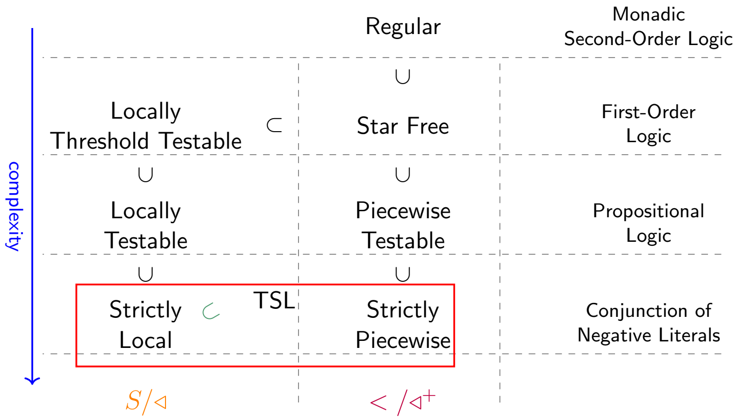
Efficient learning of ITSL **online**!



- **Learnable?** Jardine & Heinz (2016), Jardine & McMullin (2017), McMullin et al. (2019), Lambert (2021) a.o.

Overview

Subregular Languages[1]



[1] McNaughton & Papert (1976), Heinz (2011), Chandlee & Heinz (2014), De Santo & Graf (2019), De Santo & Rawski (2022), a.o.

Unbounded Dependencies as TSL

- **Ineseño Chumash Sibilant Harmony**
  Sibilants must not disagree in anteriority.
  (Applegate 1972)

  (1) a. * ha**s**xintilawaʃ
  b. * haʃxintilawa**s**
  c. haʃxintilawaʃ

- What do we need to project? [+strident]
- What do we need to ban? *[+ant][−ant],*[−ant][+ant]
  I.E. *sʃ, *sʒ, *zʃ, *zʒ, *ʃs, *ʒs, *ʃz, *ʒz

Example: TSL Ineseño Chumash

s ʃ	ʃ ʃ
* \$ h a s x i n t i l a w ʃ \$	ok \$ h a ʃ x i n t i l a w ʃ \$

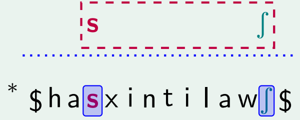
Unbounded Dependencies as TSL

- **Ineseño Chumash Sibilant Harmony**
  Sibilants must not disagree in anteriority.
  (Applegate 1972)

(1) a. * hasxintilawaʃ
    b. * haʃxintilawas
    c. haʃxintilawaʃ

- What do we need to project? [+strident]
- What do we need to ban? *[+ant][−ant],*[−ant][+ant]
  I.E. *sʃ, *sʒ, *zʃ, *zʒ, *ʃs, *ʒs, *ʃz, *ʒz

Example: TSL Ineseño Chumash



Ineseño Chumash: The Full Picture

Sibilant Harmony in Ineseño Chumash (McMullin 2016)

1) Unbounded sibilant harmony

a.	/k-su-ʃojin/	kʃuʃojin	"I darken it"
b.	/k-su-k'ili-mekeken-ʃ/	kʃuk'ilimekeketʃ	"I straighten up"

2) /s/ → [ʃ] when preceding (adjacent) [t, n, l]

a.	/s-lok'in/	ʃlok'in	"he cuts it"
b.	/s-tepuʔ/	ʃtepuʔ	"he gambles"

3) Long-distance agreement overrides local disagreement

a.	/s-iʃt-iʃti-jep-us/	sististijepus	"they show him"
b.	/s-net-us/	snetus	"he does it to him"

Ineseño Chumash: The Full Picture

Sibilant Harmony in Ineseño Chumash (McMullin 2016)

1) Unbounded sibilant harmony

a. /k-su-ʃojin/	kʃuʃojin	"I darken it"
b. /k-su-k'ili-mekeken-ʃ/	kʃuk'ilimekeketʃ	"I straighten up"

2) /s/ → [ʃ] when preceding (adjacent) [t, n, l]

a. /s-lok'in/	ʃlok'in	"he cuts it"
b. /s-tepuʔ/	ʃtepuʔ	"he gambles"

3) Long-distance agreement overrides local disagreement

a. /s-iʃt-iʃti-jep-us/	sististijepus	"they show him"
b. /s-net-us/	snetus	"he does it to him"

Ineseño Chumash: The Full Picture

Sibilant Harmony in Ineseño Chumash (McMullin 2016)

1) Unbounded sibilant harmony

a.	/k-su-ʃojin/	kʃuʃojin	"I darken it"
b.	/k-su-k'ili-mekeken-ʃ/	kʃuk'ilimekeketʃ	"I straighten up"

2) /s/→ [ʃ] when preceding (adjacent) [t, n, l]

a.	/s-lok'in/	ʃlok'in	"he cuts it"
b.	/s-tepuʔ/	ʃtepuʔ	"he gambles"

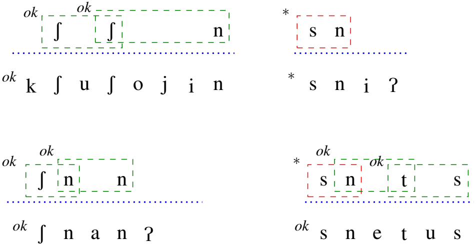
3) Long-distance agreement overrides local disagreement

a.	/s-iʃt-iʃti-jep-us/	sististijepus	"they show him"
b.	/s-net-us/	snetus	"he does it to him"

Ineseño Chumash is not TSL

Ineseño Chumash Sibilant Harmony (Revisited)

- anticipatory sibilant harmony [*sʃ, *s̪ʃ]
- palatalization to avoid local restriction [*sn, *st, *sl]
- sibilant harmony overrides palatalization



Input-Sensitive TSL (ITSL) Languages

TSL languages are characterized by:

- a 1-local projection function E_T
- strictly k-local constraints applied on T

ITSL (De Santo & Graf 2019)

- Tier projection controlled by:
  1. label of segment
  2. n-local context
- strictly k-local constraints applied on T

ITSL

$1 + 2$

TSL

1

An ITSL Account of Ineseño Chumash

Ineseño Chumash Sibilant Harmony (Revisited)

- anticipatory sibilant harmony [*sʃ, *sʃ]
- palatalization to avoid local restriction [*sn, *st, *sl]
- sibilant harmony overrides palatalization

$ s n e t u s $

An ITSL Account of Ineseño Chumash

Ineseño Chumash Sibilant Harmony (Revisited)

- anticipatory sibilant harmony [*sʃ, *sʃ]
- palatalization to avoid local restriction [*sn, *st, *sl]
- sibilant harmony overrides palatalization

$ s n e t u s $

An ITSL Account of Ineseño Chumash

Ineseño Chumash Sibilant Harmony (Revisited)

- anticipatory sibilant harmony [*sʃ, *sʃ]
- palatalization to avoid local restriction [*sn, *st, *sl]
- sibilant harmony overrides palatalization

s

[$ s] n e t u s $

An ITSL Account of Ineseño Chumash

Ineseño Chumash Sibilant Harmony (Revisited)

- anticipatory sibilant harmony [*sʃ, *sʃ]
- palatalization to avoid local restriction [*sn, *st, *sl]
- sibilant harmony overrides palatalization

s n

$ s n e t u s $

An ITSL Account of Ineseño Chumash

Ineseño Chumash Sibilant Harmony (Revisited)

- anticipatory sibilant harmony [*sʃ, *sʃ]
- palatalization to avoid local restriction [*sn, *st, *sl]
- sibilant harmony overrides palatalization

s n

$ s n e t u s $

An ITSL Account of Ineseño Chumash

Ineseño Chumash Sibilant Harmony (Revisited)

- anticipatory sibilant harmony [*sʃ, *sʃ]
- palatalization to avoid local restriction [*sn, *st, *sl]
- sibilant harmony overrides palatalization

s n

$ s n e t u s $

An ITSL Account of Ineseño Chumash

Ineseño Chumash Sibilant Harmony (Revisited)

- anticipatory sibilant harmony [*sʃ, *sʃ]
- palatalization to avoid local restriction [*sn, *st, *sl]
- sibilant harmony overrides palatalization

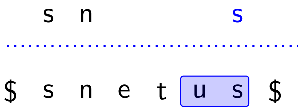
s n

$ s n e [t u] s $

An ITSL Account of Ineseño Chumash

Ineseño Chumash Sibilant Harmony (Revisited)

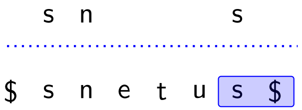
- anticipatory sibilant harmony [*sʃ, *sʃ]
- palatalization to avoid local restriction [*sn, *st, *sl]
- sibilant harmony overrides palatalization



An ITSL Account of Ineseño Chumash

INESEÑO CHUMASH Sibilant Harmony (Revisited)

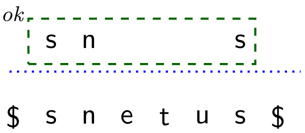
- anticipatory sibilant harmony [*sʃ, *sʃ]
- palatalization to avoid local restriction [*sn, *st, *sl]
- sibilant harmony overrides palatalization



An ITSL Account of Ineseño Chumash

INESEÑO CHUMASH Sibilant Harmony (Revisited)

- anticipatory sibilant harmony [*sʃ, *sʃ]
- palatalization to avoid local restriction [*sn, *st, *sl]
- sibilant harmony overrides palatalization



ITSL: Recap

Input-Sensitive TSL (ITSL; De Santo & Graf, 2019)

- n-local projection function
- strictly k-local constraints enforced on T.

- Natural generalization of TSL
- Covers a variety of patterns
  Korean vowel harmony, UTP, Yaka nasal harmony, ...
- Gold learnable
  Efficiently learnable?

Learning TSL and ITSL

Learning TSL_k Efficiently

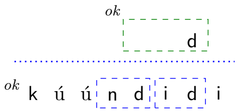
- batch learning: Jardine & Heinz (2016), Jardine & McMullin (2017)
- multiple TSL: McMullin et al. (2019)
- Incrementally: Lambert (2021)

Batch Learning ITSL_k^m Efficiently

- ITSL_2^2: De Santo & Aksenova (2021)
- multiple ITSL: De Santo & Aksenova (2021)
- evaluation: Johnson & De Santo (2023)

Learning ITSL$_k^m$ Online

- **Goal:** Efficiently learn ITSL$_k^m$ grammars from positive data, incrementally
- Efficiently: polynomial in time and data
  (Gold 1967, De la Higuera 2010)



Core Idea

- ITSL$_k^m$ $\approx$ TSL$_k$ with enriched symbols
  $\Rightarrow$ An element and its $(m-1)$-context treated as a unit
  (De Santo & Aksenova 2021)
- Lambert (2021)'s TSL strategy generalized over m-grams

Lambert (2021): Learning TSL Incrementally

A TSL_k learner must identify:

- the tier alphabet T
- k-local tier constraints

String Extension Learning (Heinz 2010)

- Stringsets described by permitted factors of length $\leq k$.
- Start with the empty grammar updated to include all factors attested in each input string.

For TSL the trick is deciding **salience**!

Lambert (2021): Learning TSL Incrementally

A TSL_k learner must identify:

- the tier alphabet T
- k-local tier constraints

String Extension Learning (Heinz 2010)

- Stringsets described by permitted factors of length $\leq k$.
- Start with the empty grammar updated to include all factors attested in each input string.

For TSL the trick is deciding **salience**!

Lambert (2021): Learning TSL Incrementally

A TSL_k learner must identify:

- the tier alphabet T
- k-local tier constraints

String Extension Learning (Heinz 2010)

- Stringsets described by permitted factors of length $\leq k$.
- Start with the empty grammar updated to include all factors attested in each input string.

For TSL the trick is deciding **salience**!

Lambert (2021) cont.

For TSL the trick is deciding **salience**!

Core Structures

An TSL_n grammar can be represented as:

- G_l: the set of attested factors of width bounded by $k+1$
- G_s: the set of augmented subsequences bounded by k

Core Idea (Lambert & Rogers 2020)

The symbols that are not both freely insertable and deletable are the salient ones!

Lambert (2021) cont.

For TSL the trick is deciding **salience**!

Core Structures

An TSL_n grammar can be represented as:

- G_l: the set of attested factors of width bounded by $k+1$
- G_s: the set of augmented subsequences bounded by k

Core Idea (Lambert & Rogers 2020)

The symbols that are not both freely insertable and deletable are the salient ones!

Salience, Tier Membership, and Factors

A TSL_n grammar can be represented as:

- G_l: the set of attested factors of width bounded by $k+1$
- G_s: the set of augmented subsequences bounded by k

Collect the set of attested factors! Given `cabacba`:

$$\{\varepsilon, \texttt{a}, \texttt{b}, \texttt{c}, \texttt{ab}, \texttt{ac}, \texttt{ba}, \texttt{ca}, \texttt{cb}, \texttt{aba}, \texttt{acb}, \texttt{bac}, \texttt{cab}, \texttt{cba}\}$$

Then $\Rightarrow$ tier constraints are factors of salient symbols over relativized adjacency!

Salience, Tier Membership, and Factors

A TSL_n grammar can be represented as:

- G_l: the set of attested factors of width bounded by $k + 1$
- G_s: the set of augmented subsequences bounded by k

Collect the set of attested factors! Given cabacba:

$$\{\varepsilon, \texttt{a}, \texttt{b}, \texttt{c}, \texttt{ab}, \texttt{ac}, \texttt{ba}, \texttt{ca}, \texttt{cb}, \texttt{aba}, \texttt{acb}, \texttt{bac}, \texttt{cab}, \texttt{cba}\}$$

Then $\Rightarrow$ tier constraints are factors of salient symbols over relativized adjacency!

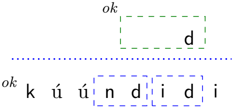
Salience, Tier Membership, and Factors [cont.]

An TSL_n grammar can be represented as:

- G_l: the set of attested factors of width bounded by $k+1$
- G_s: the set of augmented subsequences bounded by k
- Given cabacba:

Subsequence	Intervener Sets
ε	$\{\{\}\}$
a	$\{\{\}\}$
b	$\{\{\}\}$
c	$\{\{\}\}$
aa	$\{\{b\},\{b,c\}\}$
ab	$\{\{\},\{c\}\}$
ac	$\{\{\}\}$
ba	$\{\{\}\}$
bb	$\{\{a,c\}\}$
bc	$\{\{a\}\}$
ca	$\{\{\},\{b\}\}$
cb	$\{\{\},\{a\}\}$
cc	$\{\{a,b\}\}$

From TSL_n to ITSL_k^m



De Santo & Aksenova (2021)

A unary symbol and its $(m-1)$-context treated as a single element in the ITSL alphabet!

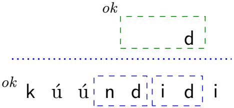
The string cabacba is represented as:

$$\langle \text{ca}, \text{ab}, \text{ba}, \text{ac}, \text{cb}, \text{ba} \rangle$$

The set of attested factors for $m = 2$ becomes:

$$\{\langle\rangle, \langle \text{ab}\rangle, \langle \text{ac}\rangle, \langle \text{ba}\rangle, \langle \text{ca}\rangle, \langle \text{cb}\rangle, \langle \text{ab}, \text{ba}\rangle, \langle \text{ac}, \text{cb}\rangle, \langle \text{ba}, \text{ac}\rangle, \langle \text{ca}, \text{ab}\rangle, \langle \text{cb}, \text{ba}\rangle, \langle \text{ab}, \text{ba}, \text{ac}\rangle, \langle \text{ac}, \text{cb}, \text{ba}\rangle, \langle \text{ba}, \text{ac}, \text{cb}\rangle, \langle \text{ca}, \text{ab}, \text{ba}\rangle\}$$

From TSL_n to ITSL_k^m



De Santo & Aksenova (2021)

A unary symbol and its $(m-1)$-context treated as a single element in the ITSL alphabet!

The string cabacba is represented as:

$$\langle \texttt{ca}, \texttt{ab}, \texttt{ba}, \texttt{ac}, \texttt{cb}, \texttt{ba} \rangle$$

The set of attested factors for $m = 2$ becomes:

$$\{\langle\rangle, \langle \texttt{ab} \rangle, \langle \texttt{ac} \rangle, \langle \texttt{ba} \rangle, \langle \texttt{ca} \rangle, \langle \texttt{cb} \rangle, \langle \texttt{ab}, \texttt{ba} \rangle, \langle \texttt{ac}, \texttt{cb} \rangle, \langle \texttt{ba}, \texttt{ac} \rangle, \langle \texttt{ca}, \texttt{ab} \rangle, \langle \texttt{cb}, \texttt{ba} \rangle, \langle \texttt{ab}, \texttt{ba}, \texttt{ac} \rangle, \langle \texttt{ac}, \texttt{cb}, \texttt{ba} \rangle, \langle \texttt{ba}, \texttt{ac}, \texttt{cb} \rangle, \langle \texttt{ca}, \texttt{ab}, \texttt{ba} \rangle\}$$

Augmented Subsequences in TSL_2 vs. $ITSL_2^2$

Subsequence	Intervener Sets
ε	$\{\{\}\}$
a	$\{\{\}\}$
b	$\{\{\}\}$
c	$\{\{\}\}$
aa	$\{\{b\},\{b,c\}\}$
ab	$\{\{\},\{c\}\}$
ac	$\{\{\}\}$
ba	$\{\{\}\}$
bb	$\{\{a,c\}\}$
bc	$\{\{a\}\}$
ca	$\{\{\},\{b\}\}$
cb	$\{\{\},\{a\}\}$
cc	$\{\{a,b\}\}$

Subsequence	Intervener Sets
$\langle\rangle$	$\{\{\}\}$
$\langle$ab$\rangle$	$\{\{\}\}$
$\langle$ac$\rangle$	$\{\{\}\}$
$\langle$ba$\rangle$	$\{\{\}\}$
$\langle$ca$\rangle$	$\{\{\}\}$
$\langle$cb$\rangle$	$\{\{\}\}$
$\langle$ab, ac$\rangle$	$\{\{$ba$\}\}$
$\langle$ab, ba$\rangle$	$\{\{\}\}$
$\langle$ab, cb$\rangle$	$\{\{$ac, ba$\}\}$
$\langle$ac, ba$\rangle$	$\{\{$cb$\}\}$
$\langle$ac, cb$\rangle$	$\{\{\}\}$
$\langle$ba, ac$\rangle$	$\{\{\}\}$
$\langle$ba, ba$\rangle$	$\{\{$ac, cb$\}\}$
$\langle$ba, cb$\rangle$	$\{\{$ac$\}\}$
$\langle$ca, ab$\rangle$	$\{\{\}\}$
$\langle$ca, ac$\rangle$	$\{\{$ab, ba$\}\}$
$\langle$ca, ba$\rangle$	$\{\{$ab$\}\}$
$\langle$ca, cb$\rangle$	$\{\{$ab, ac, ba$\}\}$
$\langle$cb, ba$\rangle$	$\{\{\}\}$

Learning ITSL$_k$ Online

An ITSL$_k^m$ grammar can be represented as:

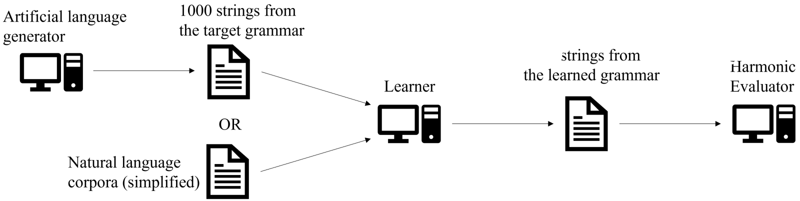
- G_l: the set of attested factors of width bounded by $k+1$
- G_s: the set of augmented subsequences bounded by k

Core Idea

- A symbol and its $(m-1)$-context treated as a unit
- Lambert (2021)'s TSL strategy generalized over m-grams

- Linear wrt input size:
  $\mathcal{O}(nk\log(|\Sigma|^m))$ time and $\mathcal{O}((|\Sigma|^m)^{k+1})$ space!
- Provably correct!
  Assumption: Input sample is *characteristic*
  $\Rightarrow$ what happens with realistic datasets?

Evaluation (Aksenova 2020)



- TSL and ITSL mplemented in Python 3 following requirements of SigmaPie
- Artificial datasets exemplifying different subregular classes
- 3 simplified natural language corpora (German, Finnish, Turkish)
- Proportion of first 5000 strings accepted by the learned grammar also accepted by the target grammar
- Learning + evaluation iterated 10 times

Evaluation

	TSL	ITSL
Word-final devoicing		
T	✓	✓
A	100%	100%
N_G	100%	100%
Single vowel harmony without blocking		
T	✓	✓
A	100%	100%
N_F	100%	100%
Single vowel harmony with blocking		
T	✓	✓
A	100%	100%
Several vowel harmonies without blocking		
T	✓	✓
A	100%	100%
Several vowel harmonies with blocking		
T	✓	✓
A	100%	100%
N_T	100%	100%

Evaluation

	TSL	ITSL
Word-final devoicing		
T	✓	✓
A	100%	100%
N_G	100%	100%
Single vowel harmony without blocking		
T	✓	✓
A	100%	100%
N_F	100%	100%
Single vowel harmony with blocking		
T	✓	✓
A	100%	100%
Several vowel harmonies without blocking		
T	✓	✓
A	100%	100%
Several vowel harmonies with blocking		
T	✓	✓
A	100%	100%
N_T	100%	100%

	TSL	ITSL
Unbounded tone plateauing		
T	✗	✓
A	9.97% (0.51%)	100%
First-Last Assimilation		
T	✗	✓
A	50.02%	100%
Locally-driven long-distance assimilation (ITSL restriction)		
T	✗	✓
A	94.88% (0.15%)	100%

Conclusion

In this paper

- Efficiently learn ITSL_k^n grammars incrementally from positive data
- No a-priori information on the content of tiers or the constraints
  - Intuitive extension of existing results
  - Python implementation available
  - Importance of evaluation pipelines!

Future Work

- Further testing on artificial and corpus data
- Extension to multiple-ITSL grammars
- Additional variations: Stochastic variants, natural classes for contexts, etc.

Conclusion

In this paper

- Efficiently learn ITSL_k^n grammars incrementally from positive data
- No a-priori information on the content of tiers or the constraints
  - Intuitive extension of existing results
  - Python implementation available
  - Importance of evaluation pipelines!

Future Work

- Further testing on artificial and corpus data
- Extension to multiple-ITSL grammars
- Additional variations: Stochastic variants, natural classes for contexts, etc.

Thank you!

Get the paper!

Code and Data!

Appendix

Locality Over Tiers

* $ s t a j a n o w o n w a ʃ $

- Sibilants can be arbitrarily far away from each other!
- **Problem**: Domains of size n miss teh crucial generalization!

Tier-based Strictly Local (TSL) Grammars (?)

- Projection of selected segments on a tier T (Goldsmith 1976)
- Strictly local constraints over T determine wellformedness
- Unbounded dependencies are local over **tiers**

Locality Over Tiers

*$ **s** t a j a n o w o n w a ʃ $

- Sibilants can be arbitrarily far away from each other!
- **Problem**: Domains of size n miss teh crucial generalization!

Tier-based Strictly Local (TSL) Grammars (?)

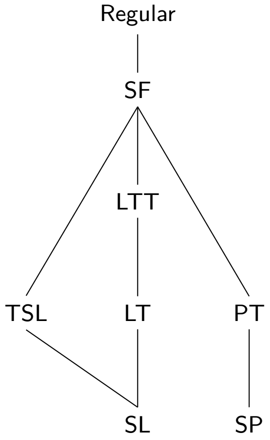
- Projection of selected segments on a tier T (Goldsmith 1976)
- Strictly local constraints over T determine wellformedness
- Unbounded dependencies are local over **tiers**

TSL: So What?

> *Descriptive characterizations focus on the* ***nature of the information*** *[...] that is needed in order to distinguish [...] a pattern*
>
> *Rogers & Pullum (2011)*

Minimal computational requirements!

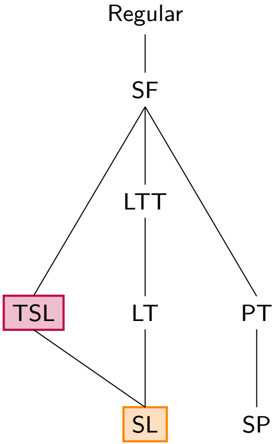
- TSL: relativized adjancency



- **But** typological variation is complex...
  (McMullin 2016, Mayer & Major 2018, De Santo & Graf 2019)

TSL: So What?

> *Descriptive characterizations focus on the* ***nature of the information*** *[...] that is needed in order to distinguish [...] a pattern*
>
> *Rogers & Pullum (2011)*

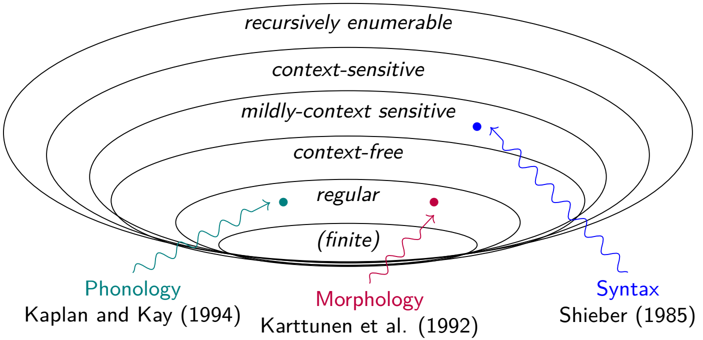


Minimal computational requirements!

- TSL: relativized adjancency

- **But** typological variation is complex...
  (McMullin 2016, Mayer & Major 2018, De Santo & Graf 2019)

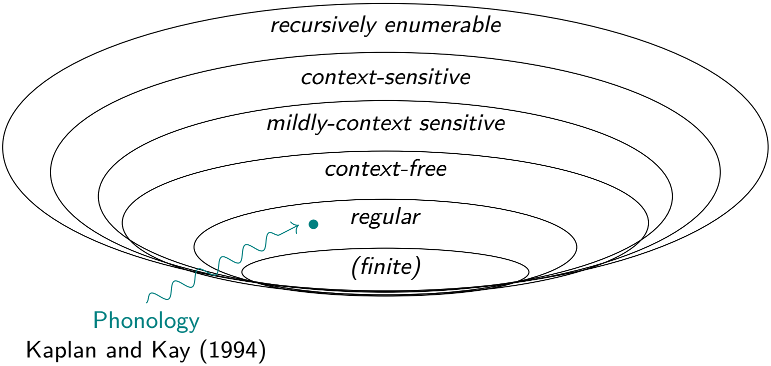
String Languages and the Chomsky-Schutzeberg Hierarchy



Precise predictions for:

- typology → e.g. no center embedding in phonology
- learnability → e.g. no Gold learning for regular languages
- cognition → e.g. finitely bounded working memory

Spoken Languages' Phonology as a Regular System



Local Phonotactic Dependencies

1. **Word-final devoicing**
   Forbid voiced segments at the end of a word

   (2) a. * rad
   b. rat

1. **Intervocalic voicing**
   Forbid voiceless segments in between two vowels

   (3) a. * faser
   b. fazer

These patters can be described by **strictly local** (SL) constraints.

Local Phonotactic Dependencies

1. **Word-final devoicing**

   Forbid voiced segments at the end of a word

   (2) a. * rad

   b. rat

1. **Intervocalic voicing**

   Forbid voiceless segments in between two vowels

   (3) a. * faser

   b. fazer

These patters can be described by **strictly local** (SL) constraints.

Local Dependencies in Phonology are SL

Example: Word-final devoicing

- Forbid voiced segments at the end of a word: *[+voice]$
- **German**: ***z**$, ***v**$,***d**$ ($ = word edge).

$ r a **d** $ $ r a t $

Example: Intervocalic voicing

- Forbid voicess segments in-between two vowels: *V[-voice]V
- **German**: *a**s**e, *i**s**e, *e**s**e, *i**s**i, ...

$ f a **s** e r $ $ f a z e r $

Local Dependencies in Phonology are SL

Example: Word-final devoicing

- Forbid voiced segments at the end of a word: *[+voice]$
- **German**: ***z**$, ***v**$,***d**$ ($ = word edge).

* $ r a **d** $ *ok* $ r a t $

Example: Intervocalic voicing

- Forbid voicess segments in-between two vowels: *V[-voice]V
- **German**: *a**s**e, *i**s**e, *e**s**e, *i**s**i, . . .

$ f a **s** e r $ $ f a z e r $

Local Dependencies in Phonology are SL

Example: Word-final devoicing

- Forbid voiced segments at the end of a word: *[+voice]$
- **German**: ***z**$, ***v**$,***d**$ ($ = word edge).

* $ r a [**d** $]   *ok* $ r a [t $]

Example: Intervocalic voicing

- Forbid voicess segments in-between two vowels: *V[-voice]V
- **German**: *a**s**e, *i**s**e, *e**s**e, *i**s**i, . . .

* $ f [a **s** e] r $ *ok* $ f [a z e] r $

Unbounded Dependencies as TSL

- **Ineseño Chumash Sibilant Harmony**
  Sibilants must not disagree in anteriority.
  (Applegate 1972)

(4) a. * ha**s**xintilawaʃ

b. * haʃxintilawa**s**

c. haʃxintilawaʃ

Ineseño Chumash with local constraints?

* $ h a **s** x i n t i l a w a ʃ $

$ h a ʃ x i n t i l a w a ʃ $

Unbounded Dependencies as TSL

- **Ineseño Chumash Sibilant Harmony**
  Sibilants must not disagree in anteriority.
  (Applegate 1972)

(4) a. * hasxintilawaʃ
b. * haʃxintilawas
c. haʃxintilawaʃ

Ineseño Chumash with local constraints?

* $ h a s x i n t i l a w a ʃ $

$ h a ʃ x i n t i l a w a ʃ $

Unbounded Dependencies as TSL

- **Ineseño Chumash Sibilant Harmony**
  Sibilants must not disagree in anteriority.
  (Applegate 1972)

(4)
a. * hasxintilawaʃ
b. * haʃxintilawas
c. haʃxintilawaʃ

Ineseño Chumash with local constraints?

* $ h a s x i n t i l a w a ʃ $

$ h a ʃ x i n t i l a w a ʃ $

Unbounded Dependencies as TSL

- **Ineseño Chumash Sibilant Harmony**
  Sibilants must not disagree in anteriority.
  (Applegate 1972)

(4)
a. * ha**s**xintilawaʃ
b. * haʃxintilawa**s**
c. haʃxintilawaʃ

Ineseño Chumash with local constraints?

* $ h a **s** x i n t i l a w a ʃ $

$ h a ʃ x i n t i l a w a ʃ $

Unbounded Dependencies as TSL

- **Ineseño Chumash Sibilant Harmony**
  Sibilants must not disagree in anteriority.
  (Applegate 1972)

(4) a. * hasxintilawaʃ
b. * haʃxintilawas
c. haʃxintilawaʃ

Ineseño Chumash with local constraints?

* $ h a s x i n t i l a w a ʃ $

$ h a ʃ x i n t i l a w a ʃ $

- **But:** Sibilants can be arbitrarily far away from each other!

* $ s t a j a n o w o n w a ʃ $

Unbounded Dependencies as TSL

- **Ineseño Chumash Sibilant Harmony**
  Sibilants must not disagree in anteriority.
  (Applegate 1972)

(4)
a. * hasxintilawaʃ
b. * haʃxintilawas
c. haʃxintilawaʃ

Ineseño Chumash with local constraints?

* $ h a s x i n t i l a w a ʃ $

$ h a ʃ x i n t i l a w a ʃ $

- **But:** Sibilants can be arbitrarily far away from each other!

* $ s t a j a n o w o n w a ʃ $

Nasal Harmony in Yaka (Hyman, 1995; Walker, 2000)

(5) a. yán-ini ‘*to cry out*’

b. yád-idi ‘*to spread*’

c. *yán-idi

(6) a. hámúk-ini ‘*to give away*’

b. míítuk-ini ‘*to sulk*’

(7) a. bíímb-idi ‘*to embrace*’

b. kúúnd-idi ‘*to bury*’

c. nááŋg-ini ‘*to last*’

(8) a. kém-ene

b. kéb-ede

Nasal Harmony in Yaka (Hyman, 1995; Walker, 2000)

(1) a. yán-ini '*to cry out*'
b. yád-idi '*to spread*'
c. *yán-idi

(2) a. hámúk-ini '*to give away*'
b. mí́ítuk-ini '*to sulk*'

(3) a. bí́ímb-idi '*to embrace*'
b. kúúnd-idi '*to bury*'
c. nááŋg-ini '*to last*'

(4) a. kem-ene
b. keb-ede

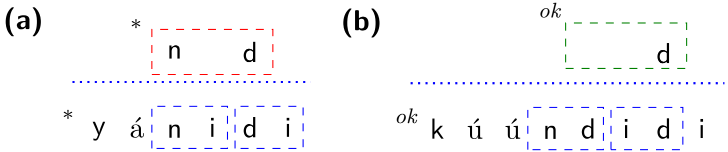
Generalization

- a nasal stop induces nasalization of voiced consonants
- nasal + voiced oral stop complexes neither trigger nor block nasality agreement
- vowel heigh harmony independent of nasalization

An (M)ITSL Account of Yaka (Part 1)

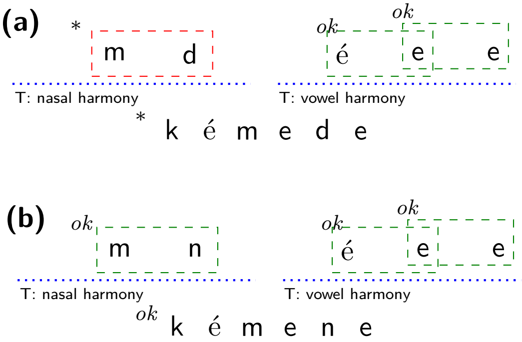
Multi Input-Sensitive TSL (MITSL; De Santo & Graf, 2019)

- **multiple** n-local projection functions E_{T_i}
- distinct strictly k-local constraints applied on each T_i.



An MITSL$_2^2$ Account of Yaka (Part 2)

- Enforce nasal harmony and vowel harmony on distinct tiers



Evaluation: Completeness

	TSL	ITSL
Word-final devoicing		
A	99.96%	71.22% (2.64%)
Single vowel harmony without blocking		
A	9.24%	7.78%
Single vowel harmony with blocking		
A	86.54%	18.64% (1.25%)
Several vowel harmonies without blocking		
A	12.64%	10.26%
Several vowel harmonies with blocking		
A	99.82%	56.90% (1.53%)
Unbounded tone plateauing		
A	99.96%	99.86%
First-Last Assimilation		
A	78.14%	73.01% (0.81%)
Locally-driven long-distance assimilation (ITSL restriction)		
A	99.96%	59.79% (1.23%)

- Learn over 1000 strings
- % 5000 strings from Σ^* accepted by target grammar also accepted by learned grammar
- Learning + Evaluation iterated 10 times

ITSL_2^2 Inference Algorithm: Example

[t!]

w	G_ℓ	G_s
	{}	{}
cabacba	{⟨⟩, ⟨ab⟩, ⟨ac⟩, ⟨ba⟩, ⟨ca⟩, ⟨cb⟩, ⟨ab, ba⟩, ⟨ac, cb⟩, ⟨ba, ac⟩, ⟨ca, ab⟩, ⟨cb, ba⟩, ⟨ab, ba, ac⟩, ⟨ac, cb, ba⟩, ⟨ba, ac, cb⟩, ⟨ca, ab, ba⟩, }.	{⟨⟨⟩, {}⟩, ⟨⟨ab⟩, {}⟩, ⟨⟨ac⟩, {}⟩, ⟨⟨ba⟩, {}⟩, ⟨⟨ca⟩, {}⟩, ⟨⟨cb⟩, {}⟩, ⟨⟨ab, ac⟩, {ba}⟩, ⟨⟨ab, ba⟩, {}⟩, ⟨⟨ab, cb⟩, {ac, ba}⟩, ⟨⟨ac, ba⟩, {cb}⟩, ⟨⟨ac, cb⟩, {}⟩, ⟨⟨ba, ac⟩, {}⟩, ⟨⟨ba, ba⟩, {ac, cb}⟩, ⟨⟨ba, cb⟩, {ac}⟩, ⟨⟨ca, ab⟩, {}⟩, ⟨⟨ca, ac⟩, {ab, ba}⟩, ⟨⟨ca, ba⟩, {ab}⟩, ⟨⟨ca, cb⟩, {ab, ac, ba}⟩, ⟨⟨cb, ba⟩, {}⟩}
abca	{⟨⟩, ⟨ab⟩, ⟨ac⟩, ⟨ba⟩, ⟨bc⟩, ⟨ca⟩, ⟨cb⟩, ⟨ab, ba⟩, ⟨ab, bc⟩, ⟨ac, cb⟩, ⟨ba, ac⟩, ⟨bc, ca⟩, ⟨ca, ab⟩, ⟨cb, ba⟩, ⟨ab, ba, ac⟩,	{⟨⟨⟩, {}⟩, ⟨⟨ab⟩, {}⟩, ⟨⟨ac⟩, {}⟩, ⟨⟨ba⟩, {}⟩, ⟨⟨bc⟩, {}⟩, ⟨⟨ca⟩, {}⟩, ⟨⟨cb⟩, {}⟩, ⟨⟨ab, ac⟩, {ba}⟩, ⟨⟨ab, ba⟩, {}⟩, ⟨⟨ab, bc⟩, {}⟩, ⟨⟨ab, ca⟩, {bc}⟩, ⟨⟨ab, cb⟩, {ac, ba}⟩, ⟨⟨ac, ba⟩, {cb}⟩,