Modeling island effects with probabilistic tier-based strictly local grammars over trees

Charles Torres[1], Kenneth Hanson[2], Thomas Graf[3], and Connor Mayer[4]

[1,4]UC Irvine and [2,3]Stony Brook University
[1]charlt4@uci.edu [2]kenneth.hanson@stonybrook.edu
[3]mail@thomasgraf.net [4]cjmayer@uci.edu

SCiL 2023
June 15–17, 2023

Gradience and Syntactic Formalisms

How does one account for gradience in grammatical acceptability judgments?

Two possibilities:

1. It's not in syntax, so don't worry
2. Extend formalisms to allow for non-discrete judgments

This work

We investigate using the second strategy with new formalisms from subregular linguistics.

Collaborators

This is joint work across two institutions: UCI and Stony Brook

Charlie Torres

Kenneth Hanson

Thomas Graf

Connor Mayer

Outline

Strictly local languages

SL-k languages are generated by grammars that **prohibit certain substrings of length** k.

Strictly local languages

SL-k languages are generated by grammars that **prohibit certain substrings of length** k.

SL-2 grammar: $G := \{\text{aa}, \text{cc}\}$

Strictly local languages

SL-k languages are generated by grammars that **prohibit certain substrings of length** k.

SL-2 grammar: $G := \{\text{aa}, \text{cc}\}$

abc *abcc*

Strictly local languages

SL-k languages are generated by grammars that **prohibit certain substrings of length** k.

SL-2 grammar: $G := \{\mathrm{aa}, \mathrm{cc}\}$

abc ✓ *abcc*

Strictly local languages

SL-k languages are generated by grammars that **prohibit certain substrings of length** k.

SL-2 grammar: $G := \{\text{aa}, \underline{\text{cc}}\}$

abc ✓ *ab*<u>*cc*</u> ✗

Long-distance dependencies

SL grammars can't capture long-distance dependencies.

Long-distance dependencies

SL grammars can't capture long-distance dependencies.

Example: Primary stress can occur anywhere, and words must contain exactly one syllable with primary stress.

Long-distance dependencies

SL grammars can't capture long-distance dependencies.

Example: Primary stress can occur anywhere, and words must contain exactly one syllable with primary stress.

$$\rtimes^{k-1} \quad \acute{\sigma} \quad \sigma^{k-1} \quad \acute{\sigma} \quad \ltimes^{k-1}$$

Long-distance dependencies

SL grammars can't capture long-distance dependencies.

Example: Primary stress can occur anywhere, and words must contain exactly one syllable with primary stress.

$$\rtimes^{k-1} \quad \acute{\sigma} \quad \sigma^{k-1} \quad \acute{\sigma} \quad \ltimes^{k-1}$$

Long-distance dependencies

SL grammars can't capture long-distance dependencies.

Example: Primary stress can occur anywhere, and words must contain exactly one syllable with primary stress.

$$\rtimes^{k-1} \; \boxed{\acute{\sigma} \quad \sigma^{k-1}} \; \acute{\sigma} \quad \ltimes^{k-1}$$

Long-distance dependencies

SL grammars can't capture long-distance dependencies.

Example: Primary stress can occur anywhere, and words must contain exactly one syllable with primary stress.

$$\rtimes^{k-1} \quad \acute{\sigma} \quad \boxed{\sigma^{k-1} \quad \acute{\sigma}} \quad \ltimes^{k-1}$$

Long-distance dependencies

SL grammars can't capture long-distance dependencies.

Example: Primary stress can occur anywhere, and words must contain exactly one syllable with primary stress.

$$\rtimes^{k-1} \quad \acute{\sigma} \quad \sigma^{k-1} \; \boxed{\acute{\sigma} \quad \ltimes^{k-1}} \; ✓$$

Long-distance dependencies

SL grammars can't capture long-distance dependencies.

Example: Primary stress can occur anywhere, and words must contain exactly one syllable with primary stress.

$$\rtimes^{k-1} \quad \acute{\sigma} \quad \sigma^{k-1} \quad \acute{\sigma} \quad \ltimes^{k-1}$$ ✓

☹

Tier-based strictly local grammars

TSL grammars **prohibit certain substrings on a tier projection.**

- Non-local dependencies become local on the tier.

Tier-based strictly local grammars

TSL grammars **prohibit certain substrings on a tier projection.**

- Non-local dependencies become local on the tier.

$T := \{\acute{\sigma}\}$ $G = \{\acute{\sigma}\acute{\sigma}, \rtimes\ltimes\}$

Tier-based strictly local grammars

TSL grammars **prohibit certain substrings on a tier projection.**

- Non-local dependencies become local on the tier.

$T := \{\acute{\sigma}\}$ $G = \{\acute{\sigma}\acute{\sigma}, \rtimes\ltimes\}$

Input: $\rtimes$ $\acute{\sigma}$ σ^* $\acute{\sigma}$ $\ltimes$

Tier-based strictly local grammars

TSL grammars **prohibit certain substrings on a tier projection.**

- Non-local dependencies become local on the tier.

$T := \{\acute{\sigma}\}$ $\qquad\qquad$ $G = \{\acute{\sigma}\acute{\sigma}, \rtimes\ltimes\}$

Tier:	$\rtimes$	$\acute{\sigma}$		$\acute{\sigma}$	$\ltimes$
Input:	$\rtimes$	$\acute{\sigma}$	σ^*	$\acute{\sigma}$	$\ltimes$

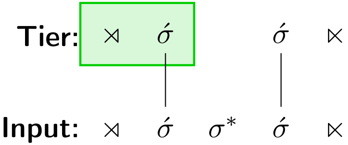
Tier-based strictly local grammars

TSL grammars **prohibit certain substrings on a tier projection.**

- Non-local dependencies become local on the tier.

$T := \{\acute{\sigma}\}$

$G = \{\acute{\sigma}\acute{\sigma}, \rtimes\ltimes\}$

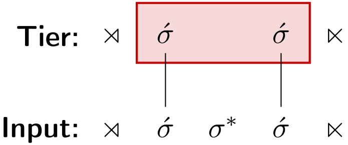


Tier-based strictly local grammars

TSL grammars **prohibit certain substrings on a tier projection.**

- Non-local dependencies become local on the tier.

$T := \{\acute{\sigma}\}$ $G = \{\underline{\acute{\sigma}\acute{\sigma}}, \rtimes\ltimes\}$

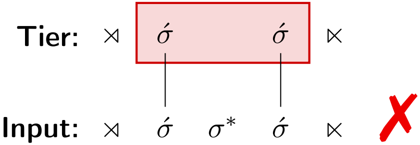


Tier-based strictly local grammars

TSL grammars **prohibit certain substrings on a tier projection.**

- Non-local dependencies become local on the tier.

$T := \{\acute{\sigma}\}$ $G = \{\underline{\acute{\sigma}\acute{\sigma}}, \rtimes\ltimes\}$

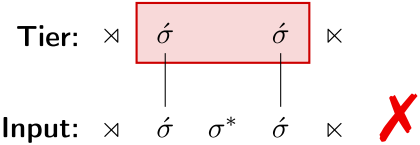


Tier-based strictly local grammars

TSL grammars **prohibit certain substrings on a tier projection.**

- Non-local dependencies become local on the tier.

$T := \{\acute{\sigma}\}$ $G = \{\underline{\acute{\sigma}\acute{\sigma}}, \rtimes\ltimes\}$



☺

Moving to non-categorical outputs

TSL grammars assign categorical membership to input strings.

Moving to non-categorical outputs

TSL grammars assign categorical membership to input strings.

- An input is either in the language or not.

Moving to non-categorical outputs

TSL grammars assign categorical membership to input strings.

- An input is either in the language or not.

Sometimes we want to model gradient properties

Moving to non-categorical outputs

TSL grammars assign categorical membership to input strings.

- An input is either in the language or not.

Sometimes we want to model gradient properties

- Acceptability ratings
  (Albright and Hayes 2003; Daland et al. 2011)

Moving to non-categorical outputs

TSL grammars assign categorical membership to input strings.

- An input is either in the language or not.

Sometimes we want to model gradient properties

- Acceptability ratings
  (Albright and Hayes 2003; Daland et al. 2011)
- Production frequencies
  (Hayes and Londe 2006; Zuraw and Hayes 2017)

Example: Uyghur backness harmony (simplified)

Description: Vowels in a word must agree in backness

- The vowel /i/ is transparent (relevant later).

TSL grammar

$G = \{$ɑæ, æɑ, uy, ...$\}$

$T = \{$æ, ɑ, ø, o, y, u$\}$

Data

pæn-lær	'science-PL'
*pæn-lɑr	
ɑt-lɑr	'horse-PL'
*ɑt-lær	

Gradient blockers

Uvulars are **gradient blockers** in Uyghur wug tests (Mayer 2021)

Production rates with no uvular:

- pæt-lær (100%) vs. pæt-lɑr (0%)

Production rates with uvular:

- pæq-lær (75%) vs. pæq-lɑr (25%)

Increases tendency for back suffixes, but not categorically!

Does /q/ project?

String: $\rtimes$ p æ q l ɑ r $\ltimes$

Does /q/ project?

Projection 1:	⋊		æ			ɑ		⋉
String:	⋊	p	æ	q	l	ɑ	r	⋉

$T = \{$æ, ɑ, ø, o, y, u$\}$

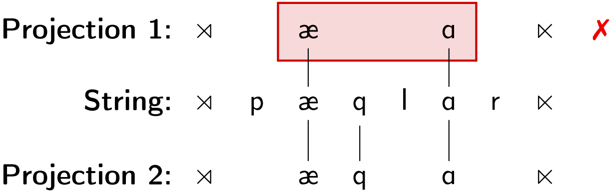
Does /q/ project?

Projection 1: ⋊ æ ɑ ⋉ ✗

String: ⋊ p æ q l ɑ r ⋉

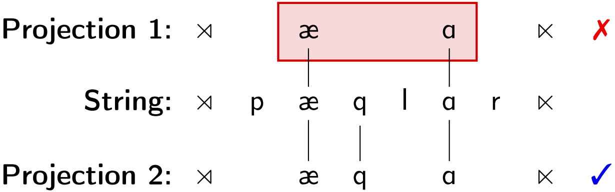
$T = \{$æ, ɑ, ø, o, y, u$\}$

Does /q/ project?



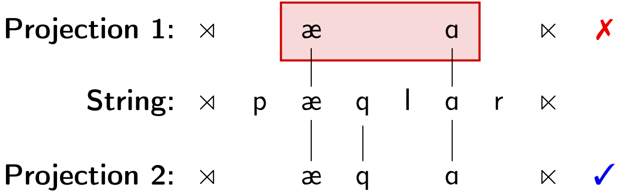
$T = \{$æ, ɑ, ø, o, y, u, $\underline{\text{q}}\}$

Does /q/ project?



$$T = \{\text{æ, ɑ, ø, o, y, u, } \underline{\text{q}}\}$$

Does /q/ project?



In either case, predicts categorical blocking or failure to block.

Probabilistic TSL

pTSL generalizes TSL by making projection *probabilistic*.

Probabilistic TSL

pTSL generalizes TSL by making projection *probabilistic*.

Each symbol has a probability of projecting, P_{proj}.

Probabilistic TSL

pTSL generalizes TSL by making projection *probabilistic*.

Each symbol has a probability of projecting, P_{proj}.

These can be used to calculate a *probability distribution* over all possible projections.

Probabilistic TSL

pTSL generalizes TSL by making projection *probabilistic*.

Each symbol has a probability of projecting, P_{proj}.

These can be used to calculate a *probability distribution* over all possible projections.

Score: Sum of the probabilities of all *grammatical* projections.

A simple case

String: ⋊ p æ q l ɑ r ⋉

P_{proj}(harmonizing vowels) $= 1$
P_{proj}(q) $= 0.25$
P_{proj}(everything else) $= 0$

A simple case

Projection 1:	$\rtimes$		æ			ɑ		$\ltimes$
			|			|		
String:	$\rtimes$	p	æ	q	l	ɑ	r	$\ltimes$

P_{proj}(harmonizing vowels) $= 1$
P_{proj}(q) $= 0.25$
P_{proj}(everything else) $= 0$

A simple case

$$1 \times 1 \times .75 \times 1 \times 1 \times 1 = 0.75$$

Projection 1:	⋊		æ			ɑ		⋉
String:	⋊	p	æ	q	l	ɑ	r	⋉

P_{proj}(harmonizing vowels) $= 1$
P_{proj}(q) $= 0.25$
P_{proj}(everything else) $= 0$

A simple case

$1 \times 1 \times .75 \times 1 \times 1 \times 1 = 0.75$

Projection 1:	$\rtimes$		æ			ɑ		$\ltimes$
String:	$\rtimes$	p	æ	q	l	ɑ	r	$\ltimes$
Projection 2:	$\rtimes$		æ	q		ɑ		$\ltimes$

P_{proj}(harmonizing vowels) $= 1$
P_{proj}(q) $= 0.25$
P_{proj}(everything else) $= 0$

A simple case

$$1 \times 1 \times .75 \times 1 \times 1 \times 1 \qquad = 0.75$$

Projection 1:	⋊		æ			ɑ		⋉
String:	⋊	p	æ	q	l	ɑ	r	⋉
Projection 2:	⋊		æ	q		ɑ		⋉

$$1 \times 1 \times .25 \times 1 \times 1 \times 1 \qquad = 0.25$$

$P_{proj}(\text{harmonizing vowels}) = 1$
$P_{proj}(\text{q}) = 0.25$
$P_{proj}(\text{everything else}) = 0$

A simple case

Projection 1: $\rtimes$ æ ɑ $\ltimes$ ✗

String: $\rtimes$ p æ q l ɑ r $\ltimes$

Projection 2: $\rtimes$ æ q ɑ $\ltimes$ ✓

$$1 \times 1 \times .25 \times 1 \times 1 \times 1 = 0.25$$

P_{proj}(harmonizing vowels) $= 1$
P_{proj}(q) $= 0.25$
P_{proj}(everything else) $= 0$

A simple case

Projection 1:	⋊		æ			ɑ		⋉	✗
String:	⋊	p	æ	q	l	ɑ	r	⋉	
Projection 2:	⋊		æ	q		ɑ		⋉	✓

$$1 \times 1 \times .25 \times 1 \times 1 \times 1 = 0.25$$

$P_{proj}(\text{harmonizing vowels}) = 1$
$P_{proj}(\text{q}) = 0.25$
$P_{proj}(\text{everything else}) = 0$

Score for /pæqlɑr/: 0.25

A more complex case

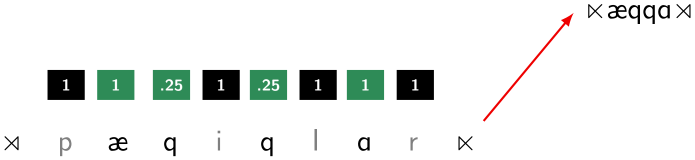
⋊ p æ q i q l ɑ r ⋉

A more complex case

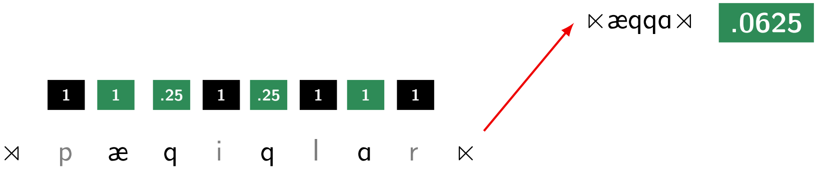
⋉ æ q q ɑ ⋊

⋊ p æ q i q l ɑ r ⋉

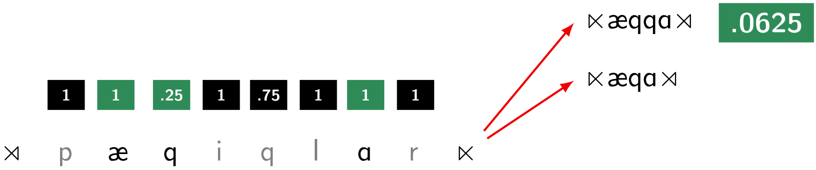
A more complex case



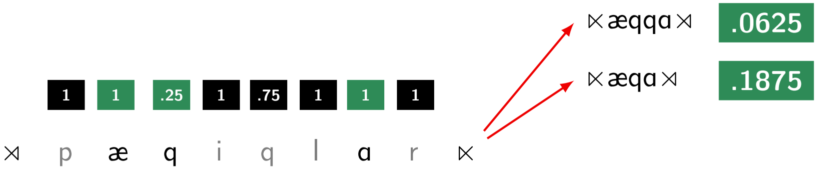
A more complex case



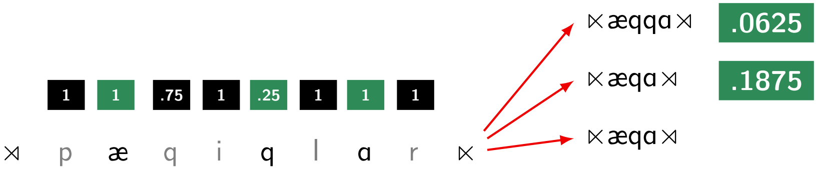
A more complex case



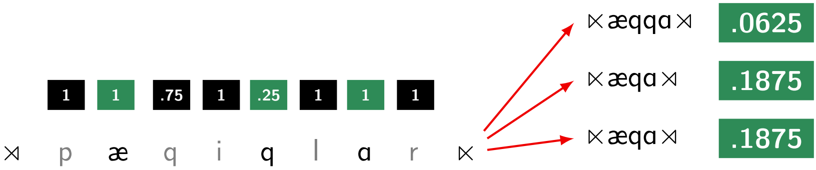
A more complex case



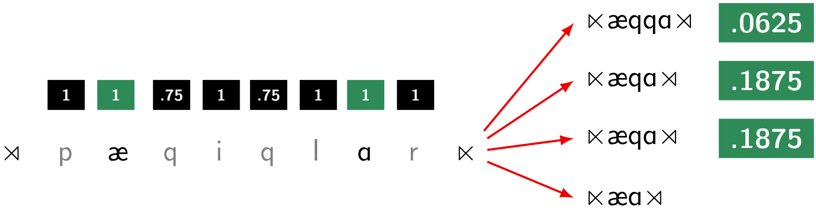
A more complex case



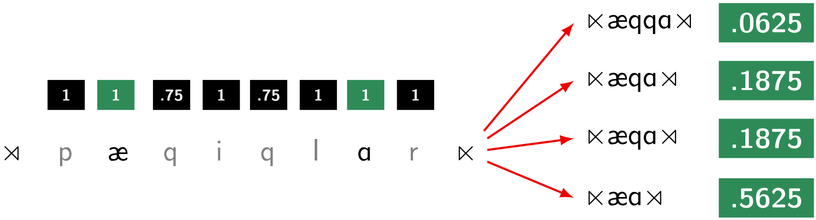
A more complex case



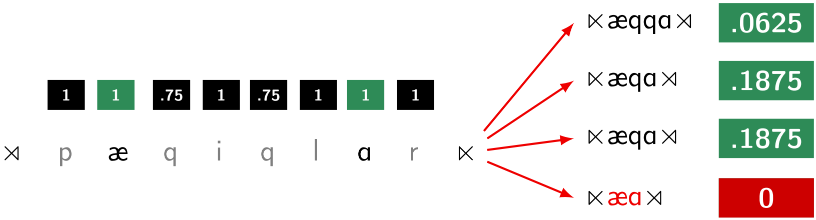
A more complex case



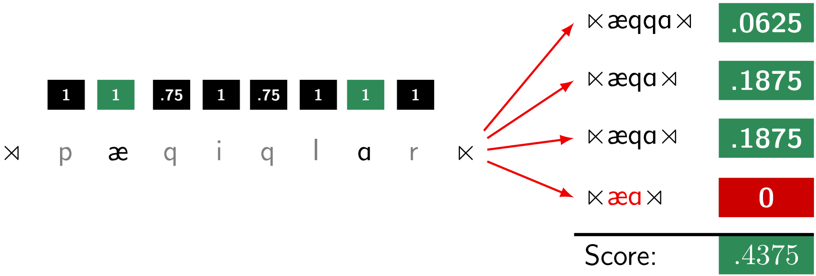
A more complex case



A more complex case



A more complex case



How does this apply to syntax?

How does this apply to syntax?

1. First, we will look at how TSL grammars over trees can regulate syntactic movement.

How does this apply to syntax?

1. First, we will look at how TSL grammars over trees can regulate syntactic movement.
2. Second, we will see if this can be extended in the same way to handle gradient syntactic judgments.

Preliminaries: Minimalist grammars and movement

Who does Mary think might buy what?

Preliminaries: Minimalist grammars and movement

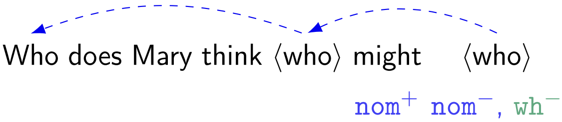
Who does Mary think ⟨who⟩ might ⟨who⟩ buy what?

Preliminaries: Minimalist grammars and movement

Who does Mary think ⟨who⟩ might ⟨who⟩ buy what?

$\texttt{nom}^-$, $\texttt{wh}^-$

Preliminaries: Minimalist grammars and movement

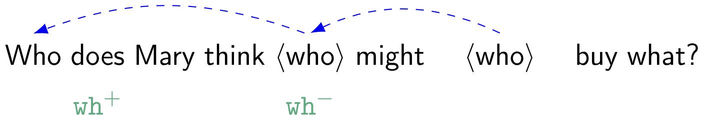


Preliminaries: Minimalist grammars and movement

Who does Mary think ⟨who⟩ might ⟨who⟩ buy what?

$\mathtt{wh}^-$

Preliminaries: Minimalist grammars and movement

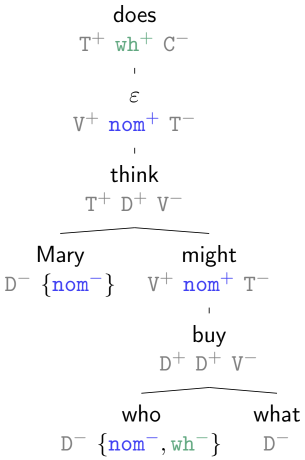


Preliminaries: Minimalist grammars and movement

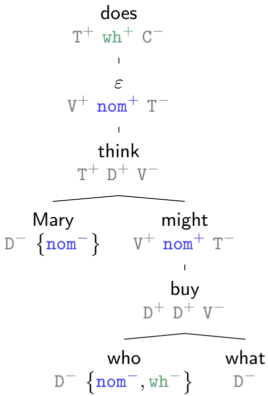
Who does Mary think ⟨who⟩ might ⟨who⟩ buy what?

Preliminaries: Minimalist grammars and movement

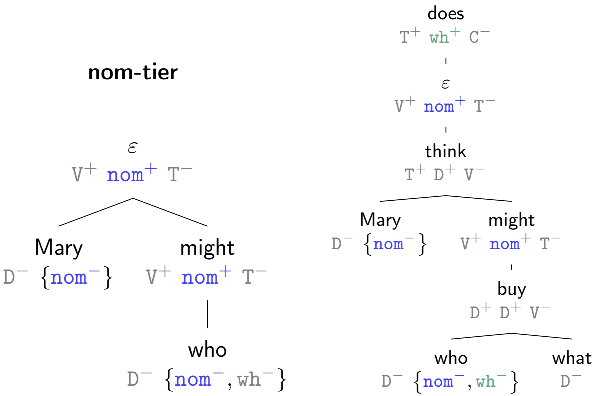
Who does Mary think ⟨who⟩ might ⟨who⟩ buy what?



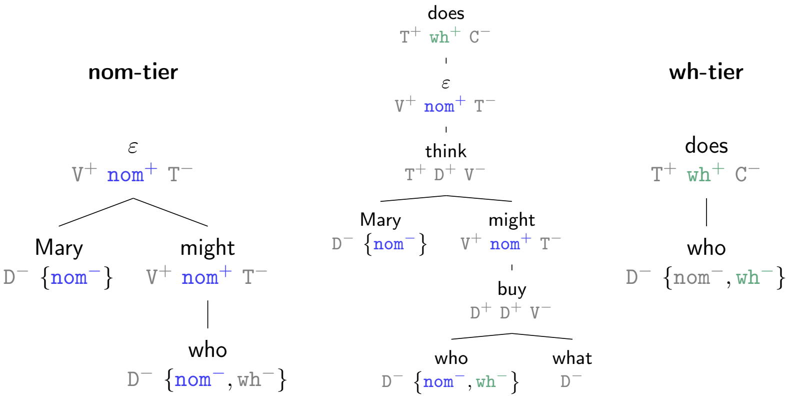
Movement in MGs is tier-based strictly local (TSL)



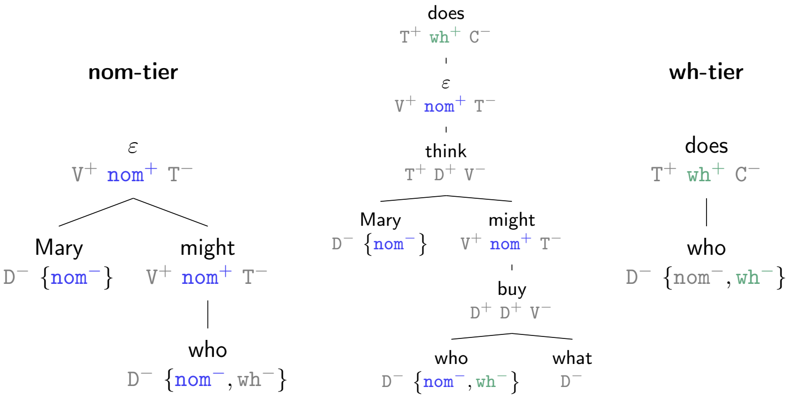
Movement in MGs is tier-based strictly local (TSL)



Movement in MGs is tier-based strictly local (TSL)



Movement in MGs is tier-based strictly local (TSL)

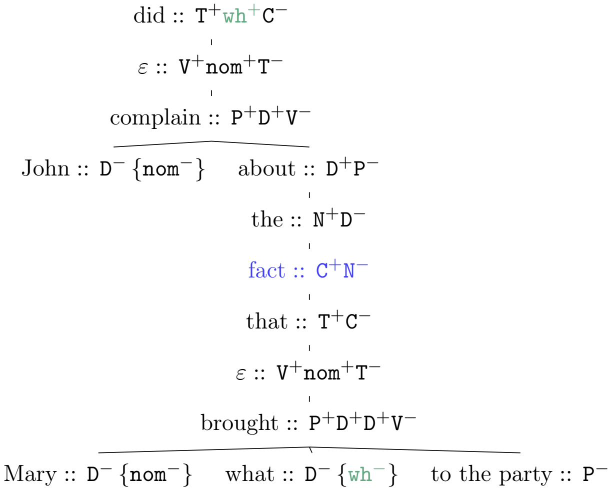


Feature checking as a constraint on each f-tier

1. Every $\texttt{f}^+$ has **exactly 1 $\texttt{f}^-$** among its daughters.
2. Every $\texttt{f}^-$ has **$\texttt{f}^+$ as its mother**.

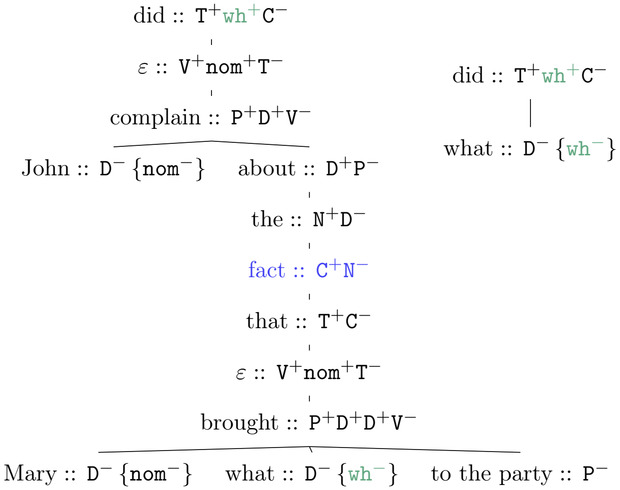
Islands as tier blockers

(1) * What did John complain about the fact that Mary brought ⟨what⟩ to the party?



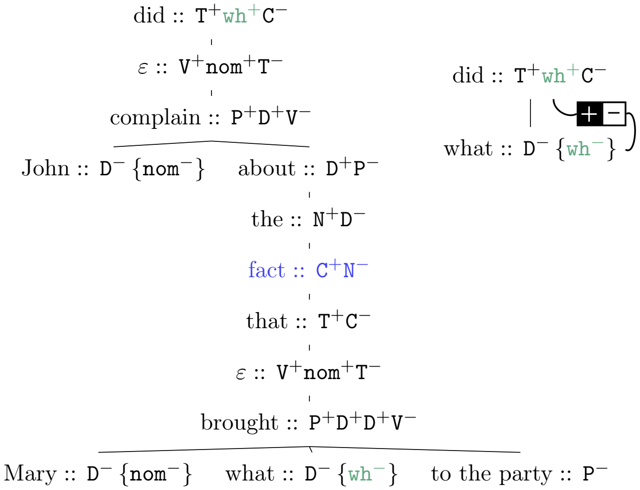
Islands as tier blockers

(1) * What did John complain about the fact that Mary brought ⟨what⟩ to the party?



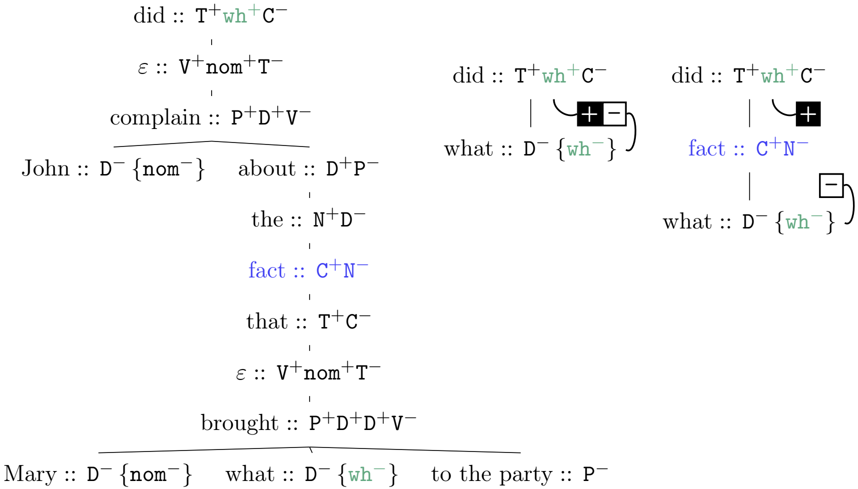
Islands as tier blockers

(1) * What did John complain about the fact that Mary brought ⟨what⟩ to the party?



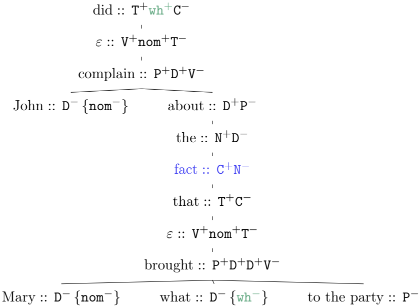
Islands as tier blockers

(1) * What did John complain about the fact that Mary brought ⟨what⟩ to the party?



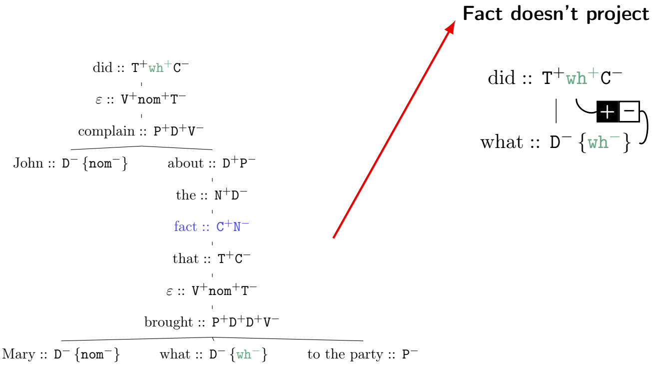
Probabilistic tree projection

- $P_{proj}(\text{fact}) = 0.7$
- $P_{proj} = 1$ for all `wh` nodes, 0 otherwise



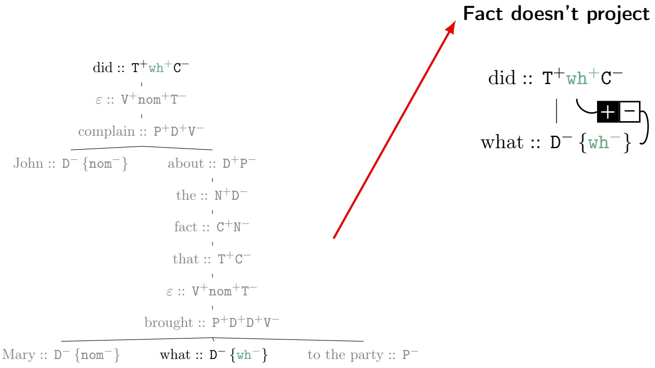
Probabilistic tree projection

- $P_{proj}(\text{fact}) = 0.7$
- $P_{proj} = 1$ for all `wh` nodes, 0 otherwise



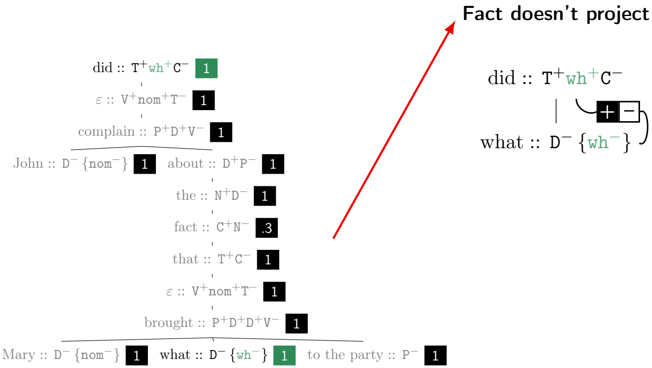
Probabilistic tree projection

- $P_{proj}(\text{fact}) = 0.7$
- $P_{proj} = 1$ for all wh nodes, 0 otherwise



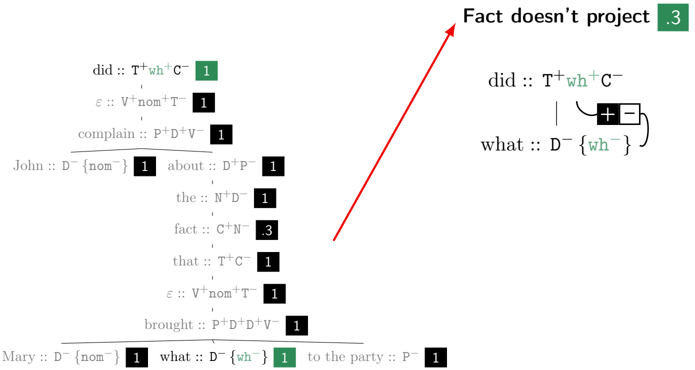
Probabilistic tree projection

- $P_{proj}(\text{fact}) = 0.7$
- $P_{proj} = 1$ for all wh nodes, 0 otherwise



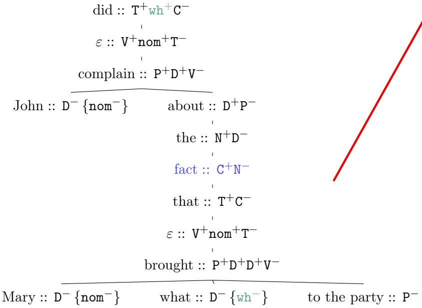
Probabilistic tree projection

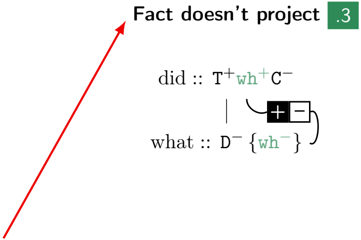
- $P_{proj}(\text{fact}) = 0.7$
- $P_{proj} = 1$ for all wh nodes, 0 otherwise



Probabilistic tree projection

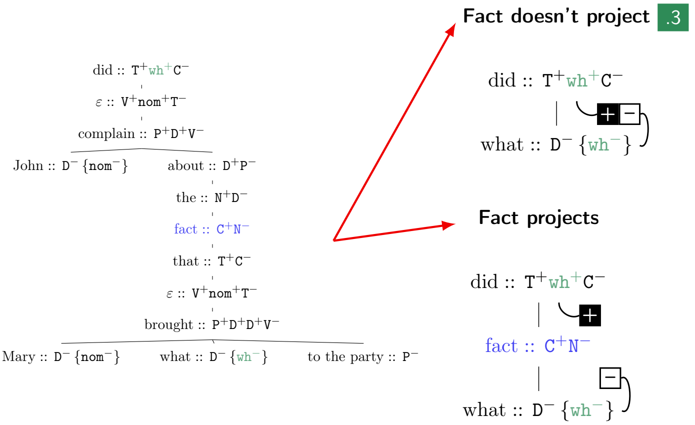
- P_{proj}(fact) $= 0.7$
- $P_{proj} = 1$ for all wh nodes, 0 otherwise





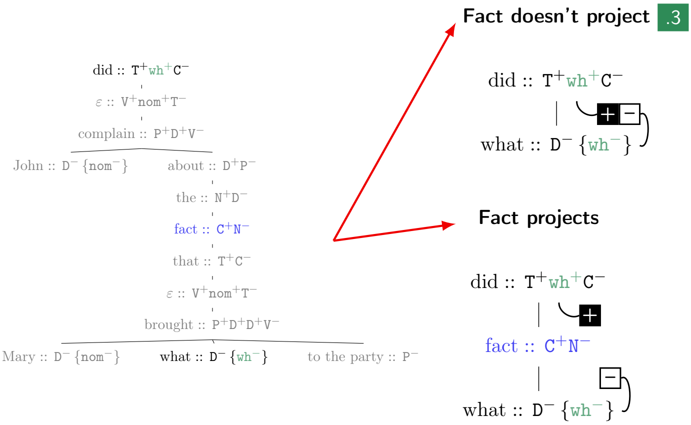
Probabilistic tree projection

- $P_{proj}(\text{fact}) = 0.7$
- $P_{proj} = 1$ for all wh nodes, 0 otherwise



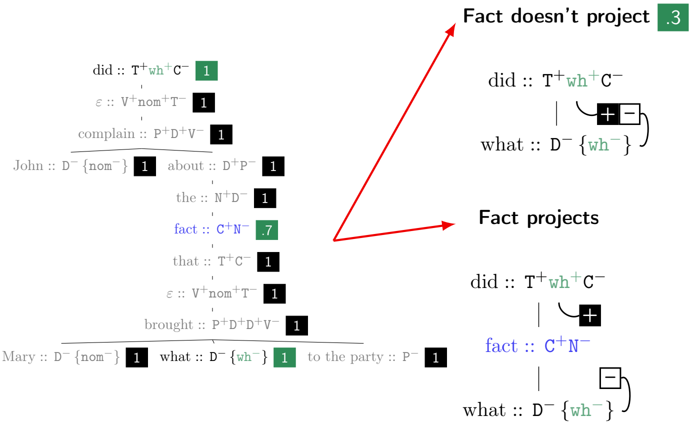
Probabilistic tree projection

- $P_{proj}(\text{fact}) = 0.7$
- $P_{proj} = 1$ for all wh nodes, 0 otherwise



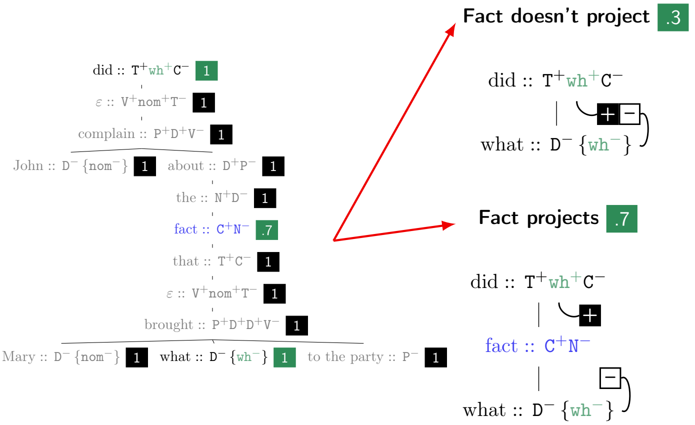
Probabilistic tree projection

- $P_{proj}(\text{fact}) = 0.7$
- $P_{proj} = 1$ for all `wh` nodes, 0 otherwise



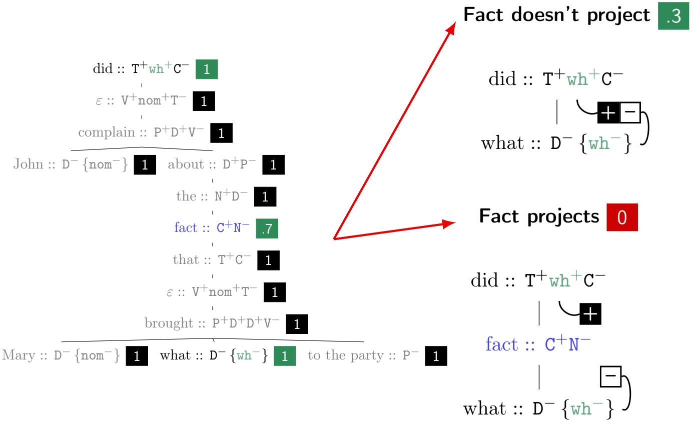
Probabilistic tree projection

- $P_{proj}(\text{fact}) = 0.7$
- $P_{proj} = 1$ for all wh nodes, 0 otherwise



Probabilistic tree projection

- $P_{proj}(\text{fact}) = 0.7$
- $P_{proj} = 1$ for all wh nodes, 0 otherwise



Modeling Island Effects using pTSL over Trees

Remainder of this talk: a modeling study that captures gradient island effects in experimental acceptability judgments.

Fit P_{proj} to English experimental data from Sprouse et al. (2016).

Figure: Jon Sprouse

The Sprouse data

The Sprouse data are sentences balanced for two factors:

The Sprouse data

The Sprouse data are sentences balanced for two factors:

- Presence of an island-effect inducing structure

The Sprouse data

The Sprouse data are sentences balanced for two factors:

- Presence of an island-effect inducing structure
- Matrix or embedded clause extraction.

The Sprouse data

The Sprouse data are sentences balanced for two factors:

- Presence of an island-effect inducing structure
- Matrix or embedded clause extraction.

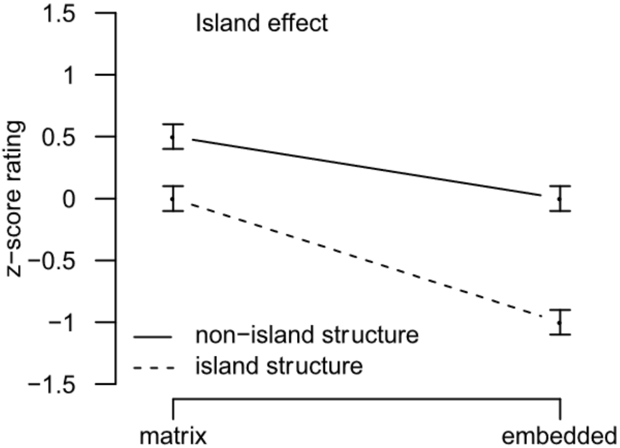
Example:

(2)
a. Who t thinks [that John bought a car]? (non-island, matrix clause)
b. What do you think [that John bought t]? (non-island, embedded clause)
c. Who t wonders [whether John bought a car]? (island, matrix clause)
d. What do you wonder [whether John bought t]? (island, embedded)

The Sprouse data

Only sentences with extraction from an embedded clause over an island structure should be ungrammatical.

- **Superadditivity**: effect of these two factors together is greater than their individual effects.



Island types

We restricted ourselves to three subsets of island effects:

1. **Whether islands**: *What do you wonder whether John bought t?
2. **Complex NP islands**: *Who did Mary deny the rumor that John likes t?
3. **Adjunct islands**: *Who did Mary complain because John likes t?

The Complex NP Constraint and Adjunct Island Constraint also apply to extraction out of relative clauses.

- We treat both as involving `wh` features for simplicity.

Adapting the data

Each of these sentences has a Likert score assigned by the participants.

Adapting the data

Each of these sentences has a Likert score assigned by the participants.

We use ratings z-score normalized by participant transformed to fall within the range $[0, 1]$.

Adapting the data

Each of these sentences has a Likert score assigned by the participants.

We use ratings z-score normalized by participant transformed to fall within the range $[0, 1]$.

We converted each sentence in the data set into dependency tree format.

Fitting the model

Some projection probabilities set *a priori*:

- Most nodes set to 0
- `wh` nodes set to 1

Free parameters: Blockers in the discrete analysis.

- $\text{that} :: \texttt{T}^{+}\texttt{C}^{-}$ (*whether/adjunct islands*)
- $\text{whether} :: \texttt{T}^{+}\texttt{C}^{-}$ (*whether/adjunct islands*)
- $\text{if} :: \texttt{T}^{+}\texttt{C}^{-}$ (*whether/adjunct islands*)
- all nodes whose feature annotation contains the substring $\texttt{C}^{+}\texttt{N}^{-}$ (*complex NP islands*)

Fitting the model

Free parameters fit to data set consisting of dependency trees with mean normalized Likert ratings.

Model assigns values in [0,1] to each tree.

We find projection probabilities that minimize the mean squared error between model and human scores.

Super-additivity: Humans vs. pTSL

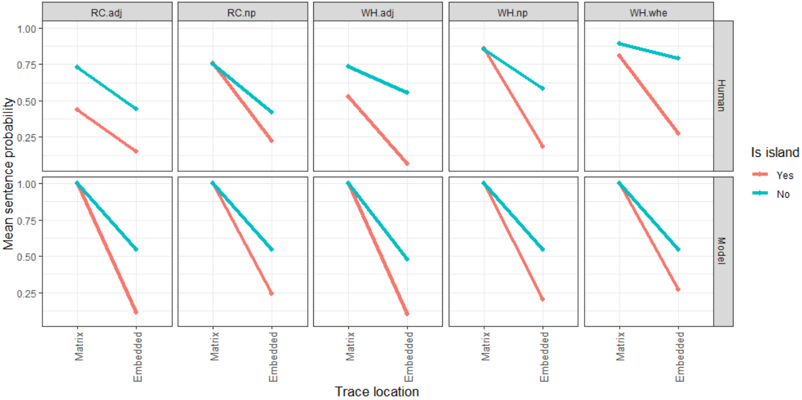


Figure: Super-additivity in a pTSL model

Fit Projection Probabilities

Node	Projection probability
that :: $\texttt{T}^{+}\texttt{C}^{-}$	.46
$\texttt{C}^{+}$ $\texttt{N}^{-}$	.63
whether :: $\texttt{T}^{+}\texttt{C}^{-}$	.73
if :: $\texttt{T}^{+}\texttt{C}^{-}$	.89

Table: Fit projection probabilities

Higher probability means greater propensity to block and induce island effects.

Discussion

The model succeeds in some respects:

- Captures supperadditivity
- Relative badness of different types of islands.

The model fails in other respects:

- Overpredicts superadditivity in some cases
- Doesn't account for variability in non-island cases.

Discussion

Many factors go into acceptability judgments:

Discussion

Many factors go into acceptability judgments:

- Syntax, processing, lexical frequency, semantics, pragmatics, information structure, etc.

Discussion

Many factors go into acceptability judgments:

- Syntax, processing, lexical frequency, semantics, pragmatics, information structure, etc.
- This model considers only syntactic effects.

Discussion

Many factors go into acceptability judgments:

- Syntax, processing, lexical frequency, semantics, pragmatics, information structure, etc.
- This model considers only syntactic effects.
- Model can be extended to account for other factors and determine which effects should be modeled as part of the grammar.

Discussion

Many factors go into acceptability judgments:

- Syntax, processing, lexical frequency, semantics, pragmatics, information structure, etc.
- This model considers only syntactic effects.
- Model can be extended to account for other factors and determine which effects should be modeled as part of the grammar.

Some of the probabilities likely encode non-syntactic features:

Discussion

Many factors go into acceptability judgments:

- Syntax, processing, lexical frequency, semantics, pragmatics, information structure, etc.
- This model considers only syntactic effects.
- Model can be extended to account for other factors and determine which effects should be modeled as part of the grammar.

Some of the probabilities likely encode non-syntactic features:

- ‘that’ isn’t typically considered a blocker.

Discussion

Many factors go into acceptability judgments:

- Syntax, processing, lexical frequency, semantics, pragmatics, information structure, etc.
- This model considers only syntactic effects.
- Model can be extended to account for other factors and determine which effects should be modeled as part of the grammar.

Some of the probabilities likely encode non-syntactic features:

- ‘that’ isn’t typically considered a blocker.
- Encodes (non-syntactic?) decrease in acceptability between matrix and embedded extraction.

Takeaways

- Converting a categorical TSL model to a probabilistic one is easy and empirically viable.

Takeaways

- Converting a categorical TSL model to a probabilistic one is easy and empirically viable.
- The same computational structure can be applied to constraints in phonology and in syntax.

Takeaways

- Converting a categorical TSL model to a probabilistic one is easy and empirically viable.
- The same computational structure can be applied to constraints in phonology and in syntax.
- pTSL over trees can capture superadditivity and gradience in syntactic island effects.

Takeaways

- Converting a categorical TSL model to a probabilistic one is easy and empirically viable.
- The same computational structure can be applied to constraints in phonology and in syntax.
- pTSL over trees can capture superadditivity and gradience in syntactic island effects.

Take-home message

pTSL over trees lets us model gradience arising from grammatical factors.

Acknowledgments

The work carried out by Kenneth Hanson and Thomas Graf as part of this project was supported by the National Science Foundation under Grant No. BCS-1845344.

References

Albright, Adam, and Bruce Hayes. 2003. Rules vs. analogy in English past tenses: a computational/experimental study. *Cognition* 90:119–161.

Daland, Robert, Bruce Hayes, James White, Marc Garellek, Andreas Davis, and Ingrid Normann. 2011. Explaining sonority projection effects. *Phonology* 28:197–234.

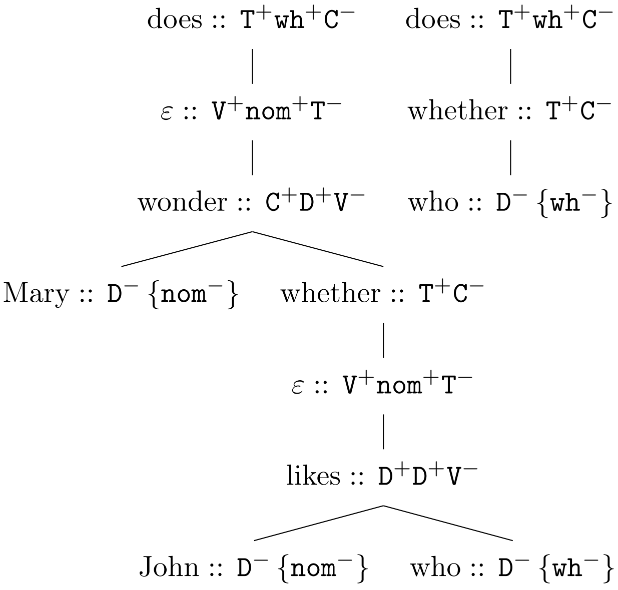
Hayes, Bruce, and Zsuzsa Londe. 2006. Stochastic phonological knowledge: the case of Hungarian vowel harmony. *Phonology* 23:59–104.

Mayer, Connor. 2021. Capturing gradience in long-distance phonology using probabilistic tier-based strictly local grammars. In *Proceedings of the Society for Computation in Linguistics 2021*, 39–50. Online: Association for Computational Linguistics. URL `https://aclanthology.org/2021.scil-1.4`.

Sprouse, Jon, Ivano Caponigro, Ciro Greco, and Carlo Cecchetto. 2016. Experimental syntax and the variation of island effects in English and Italian. *Natural Language & Linguistic Theory* 34:307–344.

Zuraw, Kie, and Bruce Hayes. 2017. Intersecting constraint families: An argument for harmonic grammar. *Language* 93:497–548.

whether island



Adjunct island