Typological Implications of an Interactive Learning Model

Coral Hughto

Manchester Phonology Meeting
26 May 2016

Background

- Traditional thought in typology: grammatical theories should generate all and only possible languages

Background

- Traditional thought in typology: grammatical theories should generate all and only possible languages
- No prediction is made about their relative frequency

Background

- Traditional thought in typology: grammatical theories should generate all and only possible languages
- No prediction is made about their relative frequency
- Some recent work combines a grammatical theory with a learning theory in order to derive predictions of typological frequency based on relative learnability (e.g. Heinz 2009, Pater & Moreton 2012, Staubs 2014)

Background

- The present (and some other) work uses Maximum Entropy (MaxEnt) grammar: weighted constraints with a probability distribution defined over output candidates

Background

- The present (and some other) work uses Maximum Entropy (MaxEnt) grammar: weighted constraints with a probability distribution defined over output candidates
- Weighted-constraint grammars have been criticized because they allow the existence of gang effects (e.g. Legendre et al. 2006)

Background

- The present (and some other) work uses Maximum Entropy (MaxEnt) grammar: weighted constraints with a probability distribution defined over output candidates
- Weighted-constraint grammars have been criticized because they allow the existence of gang effects (e.g. Legendre et al. 2006)
- That is, it's possible for multiple violations of (a) lower-weighted constraint(s) to outweigh one violation of a higher-weighted constraint

Hypothetical Gang Effect

	3 X	2 Y	H
☞A		-1	-2
B	-1		-3

	3 X	2 Y	H
C		-2	-4
☞D	-1		-3

Hypothetical Gang Effect

	3 X	2 Y	H
☞A		-1	-2
B	-1		-3

	3 X	2 Y	H
C		-2	-4
☞D	-1		-3

- One violation of constraint Y is better than one violation of constraint X

Hypothetical Gang Effect

	3 X	2 Y	H
☞A		-1	-2
B	-1		-3

	3 X	2 Y	H
C		-2	-4
☞D	-1		-3

- One violation of constraint Y is better than one violation of constraint X
- But, two violations of constraint Y is *worse* than one violation of constraint X

Background

- However, we may want this extra representational power (e.g. Staubs 2014, Pater to appear)

Background

- However, we may want this extra representational power (e.g. Staubs 2014, Pater to appear)
- Particular example: Contrast pattern between /s/ and /ʃ/ in Gujarati (Carroll 2012; citing data from Pandit 1954)

Background

- However, we may want this extra representational power (e.g. Staubs 2014, Pater to appear)
- Particular example: Contrast pattern between /s/ and /ʃ/ in Gujarati (Carroll 2012; citing data from Pandit 1954)
- Pattern can be analyzed as a gang effect, but not in standard OT (with the same constraints)

Background

- However, we may want this extra representational power (e.g. Staubs 2014, Pater to appear)
- Particular example: Contrast pattern between /s/ and /ʃ/ in Gujarati (Carroll 2012; citing data from Pandit 1954)
- Pattern can be analyzed as a gang effect, but not in standard OT (with the same constraints)
- Patterns of this type appear to be extremely rare

Background

- However, we may want this extra representational power (e.g. Staubs 2014, Pater to appear)
- Particular example: Contrast pattern between /s/ and /ʃ/ in Gujarati (Carroll 2012; citing data from Pandit 1954)
- Pattern can be analyzed as a gang effect, but not in standard OT (with the same constraints)
- Patterns of this type appear to be extremely rare
- What we want?: weighted-constraint grammar with a learning theory that predicts a bias against this pattern

Background

- However, we may want this extra representational power (e.g. Staubs 2014, Pater to appear)
- Particular example: Contrast pattern between /s/ and /ʃ/ in Gujarati (Carroll 2012; citing data from Pandit 1954)
- Pattern can be analyzed as a gang effect, but not in standard OT (with the same constraints)
- Patterns of this type appear to be extremely rare
- What we want?: weighted-constraint grammar with a learning theory that predicts a bias against this pattern
- What I show: MaxEnt + Interactive Learning Model predicts that this pattern should exist, and should be rare

Gujarati

- In Gujarati (Carroll 2012 citing Pandit 1954): /s/ and /ʃ/ contrast before front vowels, but contrast is neutralized to /s/ elsewhere

Gujarati

- In Gujarati (Carroll 2012 citing Pandit 1954): /s/ and /ʃ/ contrast before front vowels, but contrast is neutralized to /s/ elsewhere
  - /sa/ → [sa] /ʃa/ → [sa]
  - /si/ → [si] /ʃi/ → [ʃi]

Gujarati

- In Gujarati (Carroll 2012 citing Pandit 1954): /s/ and /ʃ/ contrast before front vowels, but contrast is neutralized to /s/ elsewhere
  - /sa/ → [sa] /ʃa/ → [sa]
  - /si/ → [si] /ʃi/ → [ʃi]
- This pattern can be analyzed as a gang effect using three constraints:
  - No[ʃ], No[si], IDENT[palatal]

Gujarati

- In Gujarati (Carroll 2012 citing Pandit 1954): /s/ and /ʃ/ contrast before front vowels, but contrast is neutralized to /s/ elsewhere
  - /sa/ → [sa] /ʃa/ → [sa]
  - /si/ → [si] /ʃi/ → [ʃi]
- This pattern can be analyzed as a gang effect using three constraints:
  - No[ʃ], No[si], IDENT[palatal]
- General Markedness (*[+F]), Context-specific Markedness (*[−F]/_X), Faithfulness (ID[F])

Typology

- In standard OT with ranked constraints, this generates four patterns:

1. No Variation (unmarked only)
   *[+F] >> *[−F]/__X, ID[F]
2. Full Contrast
   ID[F] >> *[+F], *[−F]/__X
3. Complementary Distribution
   *[−F]/__X >> *[+F] >> ID[F]
4. Contextual Neutralization
   *[−F]/__X >> ID[F] >> *[+F]

- Weighted constraints can generate a fifth pattern: "General-Case Neutralization" (GCN)

GCN in Gujarati

a.

/sa/	7 NO[ʃ]	6 NO[si]	4 IDENT	**H**
☞sa				0
ʃa	-1		-1	-11

b.

/ʃa/	7 NO[ʃ]	6 NO[si]	4 IDENT	**H**
☞sa			-1	-4
ʃa	-1			-7

c.

/si/	7 NO[ʃ]	6 NO[si]	4 IDENT	**H**
☞si		-1		-6
ʃi	-1		-1	-11

d.

/ʃi/	7 NO[ʃ]	6 NO[si]	4 IDENT	**H**
si		-1	-1	-10
☞ʃi	-1			-7

Observed Typological Distribution

- Carroll (2012) uses PBase (Mielke 2008) to estimate the observed typological distribution of these patterns:

Type	Observed
No Variation	44%
Full Contrast	37%
Comp. Dist.	10.3%
Contextual Neut.	8.2%
General-case Neut.	0.5%

Motivation

- To account for GCN, we need a model which predicts the existence of this pattern, and its rarity

Motivation

- To account for GCN, we need a model which predicts the existence of this pattern, and its rarity
  - Weighted-constraint grammar: predicts existence of GCN

Motivation

- To account for GCN, we need a model which predicts the existence of this pattern, and its rarity
  - Weighted-constraint grammar: predicts existence of GCN
  - Learning model: should account for low typological frequency

Motivation

- To account for GCN, we need a model which predicts the existence of this pattern, and its rarity
  - Weighted-constraint grammar: predicts existence of GCN
  - Learning model: should account for low typological frequency
- I propose:

Motivation

- To account for GCN, we need a model which predicts the existence of this pattern, and its rarity
  - Weighted-constraint grammar: predicts existence of GCN
  - Learning model: should account for low typological frequency
- I propose:
  - Use gradual, online learning algorithm

Motivation

- To account for GCN, we need a model which predicts the existence of this pattern, and its rarity
  - Weighted-constraint grammar: predicts existence of GCN
  - Learning model: should account for low typological frequency
- I propose:
  - Use gradual, online learning algorithm
  - Derive typological predictions through interaction between two learning agents (e.g. Pater & Moreton 2012, Rafferty et al. 2013)

Motivation

- To account for GCN, we need a model which predicts the existence of this pattern, and its rarity
  - Weighted-constraint grammar: predicts existence of GCN
  - Learning model: should account for low typological frequency
- I propose:
  - Use gradual, online learning algorithm
  - Derive typological predictions through interaction between two learning agents (e.g. Pater & Moreton 2012, Rafferty et al. 2013)
- Agent interaction will produce more learnable patterns more frequently

Motivation

- To account for GCN, we need a model which predicts the existence of this pattern, and its rarity
  - Weighted-constraint grammar: predicts existence of GCN
  - Learning model: should account for low typological frequency
- I propose:
  - Use gradual, online learning algorithm
  - Derive typological predictions through interaction between two learning agents (e.g. Pater & Moreton 2012, Rafferty et al. 2013)
- Agent interaction will produce more learnable patterns more frequently
- If GCN is less learnable, it should be produced less frequently by this model

The Interactive Learning Model

- Interactive learning model: two agents exchange data and between themselves generate a language (e.g. Dediu 2009, Pater & Moreton 2012)

The Interactive Learning Model

- Interactive learning model: two agents exchange data and between themselves generate a language (e.g. Dediu 2009, Pater & Moreton 2012)
- No target language; agents take turns being "teacher" and "learner"

The Interactive Learning Model

- Interactive learning model: two agents exchange data and between themselves generate a language (e.g. Dediu 2009, Pater & Moreton 2012)
- No target language; agents take turns being "teacher" and "learner"
- Contrast with an iterated learning model (e.g. Kirby & Hurford 2002) where an agent learns from a target distribution, then becomes teacher to the next agent in the learning chain

The Interactive Learning Model

- Interactive learning model: two agents exchange data and between themselves generate a language (e.g. Dediu 2009, Pater & Moreton 2012)
- No target language; agents take turns being "teacher" and "learner"
- Contrast with an iterated learning model (e.g. Kirby & Hurford 2002) where an agent learns from a target distribution, then becomes teacher to the next agent in the learning chain
  - Iterated learning models transmission of a pattern across multiple generations of speakers

The Interactive Learning Model

- Interactive learning model: two agents exchange data and between themselves generate a language (e.g. Dediu 2009, Pater & Moreton 2012)
- No target language; agents take turns being "teacher" and "learner"
- Contrast with an iterated learning model (e.g. Kirby & Hurford 2002) where an agent learns from a target distribution, then becomes teacher to the next agent in the learning chain
  - Iterated learning models transmission of a pattern across multiple generations of speakers
  - Here, modeling the probability of spontaneously generating a particular pattern in a typology

How it works

- Two agents begin equipped with a set of tableaux, a set of initial weights (zero or random)

How it works

- Two agents begin equipped with a set of tableaux, a set of initial weights (zero or random)
- One agent becomes "teacher"; it chooses an input form and samples an output according to its current grammar

How it works

- Two agents begin equipped with a set of tableaux, a set of initial weights (zero or random)
- One agent becomes "teacher"; it chooses an input form and samples an output according to its current grammar
- The "learner" samples an output for the same input according to its current grammar

How it works

- Two agents begin equipped with a set of tableaux, a set of initial weights (zero or random)
- One agent becomes "teacher"; it chooses an input form and samples an output according to its current grammar
- The "learner" samples an output for the same input according to its current grammar
- The learner compares its output to the teacher's, and if they are different, the learner updates its grammar:

How it works

- Two agents begin equipped with a set of tableaux, a set of initial weights (zero or random)
- One agent becomes "teacher"; it chooses an input form and samples an output according to its current grammar
- The "learner" samples an output for the same input according to its current grammar
- The learner compares its output to the teacher's, and if they are different, the learner updates its grammar:
- New Weights = Old Weights +
  (Teacher's Violations - Learner's Violations) * Learning Rate

How it works

- The agents take turns being teacher and learner and exchange data over some period of simulated time until they meet some predetermined stopping condition

How it works

- The agents take turns being teacher and learner and exchange data over some period of simulated time until they meet some predetermined stopping condition
- Each time the learner's output doesn't match the teacher's output, the learner updates its grammar, putting more probability on the form produced by the teacher

How it works

- The agents take turns being teacher and learner and exchange data over some period of simulated time until they meet some predetermined stopping condition
- Each time the learner's output doesn't match the teacher's output, the learner updates its grammar, putting more probability on the form produced by the teacher
- The agents can't see each others' grammars, only the outputs they produce

How it works

- The agents take turns being teacher and learner and exchange data over some period of simulated time until they meet some predetermined stopping condition
- Each time the learner's output doesn't match the teacher's output, the learner updates its grammar, putting more probability on the form produced by the teacher
- The agents can't see each others' grammars, only the outputs they produce
- Languages that the agents learn more often are predicted to occur more often typologically

Stopping Condition and Learning Rate

- Rather than use a set number of learning steps, agents interact until some minimum probability value has been assigned to all winning output candidates (here, 0.95)

Stopping Condition and Learning Rate

- Rather than use a set number of learning steps, agents interact until some minimum probability value has been assigned to all winning output candidates (here, 0.95)
- This removes any potential ambiguity in categorizing the language generated by the agents, and is possible because model tends towards accumulating probability on one output per input

Stopping Condition and Learning Rate

- Rather than use a set number of learning steps, agents interact until some minimum probability value has been assigned to all winning output candidates (here, 0.95)
- This removes any potential ambiguity in categorizing the language generated by the agents, and is possible because model tends towards accumulating probability on one output per input
- Learning rate here is 0.1

Stopping Condition and Learning Rate

- Rather than use a set number of learning steps, agents interact until some minimum probability value has been assigned to all winning output candidates (here, 0.95)
- This removes any potential ambiguity in categorizing the language generated by the agents, and is possible because model tends towards accumulating probability on one output per input
- Learning rate here is 0.1
- I did some parameter testing, and the exact value of those parameters (cutoff point and learning rate) did not seem to have a large influence on the results

Baseline

- To get a baseline comparison for estimating the model's performance, 10,000 sets of constraint weights were randomly sampled from a range 0-100

Baseline

- To get a baseline comparison for estimating the model's performance, 10,000 sets of constraint weights were randomly sampled from a range 0-100
- The distribution of these randomly sampled points provides a poor fit to the observed distribution

Baseline

- To get a baseline comparison for estimating the model's performance, 10,000 sets of constraint weights were randomly sampled from a range 0-100
- The distribution of these randomly sampled points provides a poor fit to the observed distribution
- (an r^2 value of 1.0 indicates a perfect fit)

Type	Observed	Sampling
No Variation	44%	16.8%
Full Contrast	37%	41.3%
Comp. Dist.	10.3%	8.3%
Contextual Neut.	8.2%	8.4%
General-case Neut.	0.5%	25%
r^2		0.17

Model Simulations

- In order to evaluate the model's performance, two sets of simulations were performed

Model Simulations

- In order to evaluate the model's performance, two sets of simulations were performed
  - Set 1: Agents' initial constraint weights are zero (= equal probability on all output candidates)

Model Simulations

- In order to evaluate the model's performance, two sets of simulations were performed
  - Set 1: Agents' initial constraint weights are zero (= equal probability on all output candidates)
  - Set 2: Agents' initial constraint weights are random (sampled from range 0-10)

Model Simulations

- In order to evaluate the model's performance, two sets of simulations were performed
  - Set 1: Agents' initial constraint weights are zero (= equal probability on all output candidates)
  - Set 2: Agents' initial constraint weights are random (sampled from range 0-10)
- In both sets, the model was run 1,000 times, and the resulting distribution over languages learned taken as the estimated frequency predictions

Model Simulations

- In order to evaluate the model's performance, two sets of simulations were performed
  - Set 1: Agents' initial constraint weights are zero (= equal probability on all output candidates)
  - Set 2: Agents' initial constraint weights are random (sampled from range 0-10)
- In both sets, the model was run 1,000 times, and the resulting distribution over languages learned taken as the estimated frequency predictions
- In a given run, agents interacted until one candidate in each tableau had at least 0.95 probability; learning rate = 0.1

Results

- The simulations with initial constraint weights of zero provide the best fit, though both sets are a better fit than the sampling baseline

Type	Observed	Zero	Random	Sampling
No Variation	44%	46.6%	25.7%	16.8%
Full Contrast	37%	48%	47.5%	41.3%
Comp. Dist.	10.3%	2.6%	7.7%	8.3%
Contextual Neut.	8.2%	2.7%	8%	8.4%
General-case Neut.	0.5%	0.1%	11.1%	25%
r^2		0.96	0.63	0.17

Results

- The simulations with initial constraint weights of zero provide the best fit, though both sets are a better fit than the sampling baseline

Type	Observed	Zero	Random	Sampling
No Variation	44%	46.6%	25.7%	16.8%
Full Contrast	37%	48%	47.5%	41.3%
Comp. Dist.	10.3%	2.6%	7.7%	8.3%
Contextual Neut.	8.2%	2.7%	8%	8.4%
General-case Neut.	0.5%	0.1%	11.1%	25%
r^2		0.96	0.63	0.17

- The Interactive Learning Model successfully predicts both the existence and relative rarity of the gang effect GCN pattern

Why does this work?

- The agents are essentially playing a communication game

Why does this work?

- The agents are essentially playing a communication game
- Unsuccessful communication (mismatching outputs) result in a grammar update to place more probability on the other agent's output

Why does this work?

- The agents are essentially playing a communication game
- Unsuccessful communication (mismatching outputs) result in a grammar update to place more probability on the other agent's output
- So, the agents are going to tend towards having similar grammars

Why does this work?

- The agents are essentially playing a communication game
- Unsuccessful communication (mismatching outputs) result in a grammar update to place more probability on the other agent's output
- So, the agents are going to tend towards having similar grammars
- A result of this is that the agents drift (usually fairly quickly) towards less variable grammar states, accumulating high probability on one output candidate per input (see also Hughto, Staubs, & Pater 2014)

Why does this work?

- The agents are essentially playing a communication game
- Unsuccessful communication (mismatching outputs) result in a grammar update to place more probability on the other agent's output
- So, the agents are going to tend towards having similar grammars
- A result of this is that the agents drift (usually fairly quickly) towards less variable grammar states, accumulating high probability on one output candidate per input (see also Hughto, Staubs, & Pater 2014)
- It's possible for agents to bounce out of more categorical states, but since they're very unlikely to disagree in that space (triggering a grammar update), they tend to stay there

Why does this work?

- Because agents tend to pass quickly out of more variable grammars, they also tend to end up in a grammar state relatively close to their initial state

Why does this work?

- Because agents tend to pass quickly out of more variable grammars, they also tend to end up in a grammar state relatively close to their initial state
- The constraint weights necessary to produce a categorical (at 0.95 probability) gang effect GCN pattern are relatively high compared to other patterns

Why does this work?

- Because agents tend to pass quickly out of more variable grammars, they also tend to end up in a grammar state relatively close to their initial state
- The constraint weights necessary to produce a categorical (at 0.95 probability) gang effect GCN pattern are relatively high compared to other patterns
  - A candidate's probability is proportional to the exponential of its Harmony score

Why does this work?

- Because agents tend to pass quickly out of more variable grammars, they also tend to end up in a grammar state relatively close to their initial state
- The constraint weights necessary to produce a categorical (at 0.95 probability) gang effect GCN pattern are relatively high compared to other patterns
  - A candidate's probability is proportional to the exponential of its Harmony score
  - $Prob(a) = \frac{e^{H_a}}{e^{H_a} + e^{H_b}}$

Why does this work?

- Because agents tend to pass quickly out of more variable grammars, they also tend to end up in a grammar state relatively close to their initial state
- The constraint weights necessary to produce a categorical (at 0.95 probability) gang effect GCN pattern are relatively high compared to other patterns
  - A candidate's probability is proportional to the exponential of its Harmony score
  - $Prob(a) = \frac{e^{H_a}}{e^{H_a} + e^{H_b}}$
  - Math fact: to get 0.95 probability on the winning candidate, the difference between the Harmony scores of the winning and losing candidates must be at least 3

Why does this work?

- Because agents tend to pass quickly out of more variable grammars, they also tend to end up in a grammar state relatively close to their initial state
- The constraint weights necessary to produce a categorical (at 0.95 probability) gang effect GCN pattern are relatively high compared to other patterns
  - A candidate's probability is proportional to the exponential of its Harmony score
  - $Prob(a) = \frac{e^{H_a}}{e^{H_a} + e^{H_b}}$
  - Math fact: to get 0.95 probability on the winning candidate, the difference between the Harmony scores of the winning and losing candidates must be at least 3
- GCN: min. (7, 6, 4) vs. Full Contrast: min (0, 0, 3)

General-case Neutralization

a.

/sa/	7 NO[ʃ]	6 NO[si]	4 IDENT	**H**
☞sa				0
ʃa	-1		-1	-11

b.

/ʃa/	7 NO[ʃ]	6 NO[si]	4 IDENT	**H**
☞sa			-1	-4
ʃa	-1			-7

c.

/si/	7 NO[ʃ]	6 NO[si]	4 IDENT	**H**
☞si		-1		-6
ʃi	-1		-1	-11

d.

/ʃi/	7 NO[ʃ]	6 NO[si]	4 IDENT	**H**
si		-1	-1	-10
☞ʃi	-1			-7

Full Contrast

a.

/sa/	0 NO[ʃ]	0 NO[si]	3 IDENT	**H**
☞sa				0
ʃa	-1		-1	-3

b.

/ʃa/	0 NO[ʃ]	0 NO[si]	3 IDENT	**H**
sa			-1	-3
☞ʃa	-1			0

c.

/si/	0 NO[ʃ]	0 NO[si]	3 IDENT	**H**
☞si		-1		0
ʃi	-1		-1	-3

d.

/ʃi/	0 NO[ʃ]	0 NO[si]	3 IDENT	**H**
si		-1	-1	-3
☞ʃi	-1			0

Why does this work?

1. The agents tend to quickly pass out of states of variation

Why does this work?

1. The agents tend to quickly pass out of states of variation
2. The agents tend to get stopped relatively close to their initial state

Why does this work?

1. The agents tend to quickly pass out of states of variation
2. The agents tend to get stopped relatively close to their initial state
3. The constraint weights necessary to produce the GCN pattern at 0.95 probability are higher than for other patterns

Why does this work?

1. The agents tend to quickly pass out of states of variation
2. The agents tend to get stopped relatively close to their initial state
3. The constraint weights necessary to produce the GCN pattern at 0.95 probability are higher than for other patterns

- Thus, when the initial weights are zero, agents are less likely to produce the GCN pattern than other patterns

Why does this work?

1. The agents tend to quickly pass out of states of variation
2. The agents tend to get stopped relatively close to their initial state
3. The constraint weights necessary to produce the GCN pattern at 0.95 probability are higher than for other patterns

- Thus, when the initial weights are zero, agents are less likely to produce the GCN pattern than other patterns
- When the initial constraint weights are random, there is some probability that the agents will begin in an initial state close to this pattern, and thus have a higher chance of ending there

Potential Issues

- The Interactive Learning Model does reduce the predicted probability of the gang effect GCN pattern relative to other patterns, but the strength of this result depends on the initial conditions

Potential Issues

- The Interactive Learning Model does reduce the predicted probability of the gang effect GCN pattern relative to other patterns, but the strength of this result depends on the initial conditions
  - More research is needed to explore different initial conditions and their effects on model performance, as well as their implications for a theory of human learning

Potential Issues

- The Interactive Learning Model does reduce the predicted probability of the gang effect GCN pattern relative to other patterns, but the strength of this result depends on the initial conditions
  - More research is needed to explore different initial conditions and their effects on model performance, as well as their implications for a theory of human learning
- The agents have a strong tendency to move out of variable grammar states

Potential Issues

- The Interactive Learning Model does reduce the predicted probability of the gang effect GCN pattern relative to other patterns, but the strength of this result depends on the initial conditions
  - More research is needed to explore different initial conditions and their effects on model performance, as well as their implications for a theory of human learning
- The agents have a strong tendency to move out of variable grammar states
  - However, variation does exist in natural language

Potential Issues

- The Interactive Learning Model does reduce the predicted probability of the gang effect GCN pattern relative to other patterns, but the strength of this result depends on the initial conditions
  - More research is needed to explore different initial conditions and their effects on model performance, as well as their implications for a theory of human learning
- The agents have a strong tendency to move out of variable grammar states
  - However, variation does exist in natural language
  - Variation could be learnable in this model if agents are given a memory of previously heard data

Future Work

- Currently running simulations with an Iterated Learning Model (e.g. Kirby & Hurford 2002) to compare behavior of these models

Future Work

- Currently running simulations with an Iterated Learning Model (e.g. Kirby & Hurford 2002) to compare behavior of these models
- Further exploration of model parameters, and the effects of different initial conditions, different constraints, etc.

Future Work

- Currently running simulations with an Iterated Learning Model (e.g. Kirby & Hurford 2002) to compare behavior of these models
- Further exploration of model parameters, and the effects of different initial conditions, different constraints, etc.
- Further investigation of empirical evidence for this pattern type (examples are difficult to find)

Acknowledgements

Many thanks to Joe Pater and Robert Staubs for their guidance, and to the Sound Workshop group for helpful comments and suggestions.

Empirical Note

- Finding examples of GCN-type patterns is difficult

Empirical Note

- Finding examples of GCN-type patterns is difficult
- Three constraints: General Markedness (*[+F]), Context-specific Markedness (*[−F]/_X), Faithfulness (ID[F])

Empirical Note

- Finding examples of GCN-type patterns is difficult
- Three constraints: General Markedness (*[+F]), Context-specific Markedness (*[−F]/__X), Faithfulness (ID[F])
- Marked/Unmarked contrast in specific environment; Unmarked elsewhere

Empirical Note

- Finding examples of GCN-type patterns is difficult
- Three constraints: General Markedness (*[+F]), Context-specific Markedness (*[−F]/__X), Faithfulness (ID[F])
- Marked/Unmarked contrast in specific environment; Unmarked elsewhere

/sa/	→	**[sa]**	/ʃa/	→	**[sa]**
/si/	→	**[si]**	/ʃi/	→	**[ʃi]**

Empirical Note

- Finding examples of GCN-type patterns is difficult
- Three constraints: General Markedness (*[+F]), Context-specific Markedness (*[−F]/_X), Faithfulness (ID[F])
- Marked/Unmarked contrast in specific environment; Unmarked elsewhere

/sa/	→	**[sa]**	/ʃa/	→	**[sa]**
/si/	→	**[si]**	/ʃi/	→	**[ʃi]**

- Without alternations: Marginal contrast? Segment with environment restriction?

Empirical Note

- With alternations:

/pas+a/	→	**[pasa]**	/paʃ+a/	→	**[pasa]**
/pas+i/	→	**[pasi]**	/paʃ+i/	→	**[paʃi]**

Empirical Note

- With alternations:

/pas+a/	→	**[pasa]**	/paʃ+a/	→	**[pasa]**
/pas+i/	→	**[pasi]**	/paʃ+i/	→	**[paʃi]**

- Lexically-specified process?

Empirical Note

- With alternations:

/pas+a/	→	**[pasa]**	/paʃ+a/	→	**[pasa]**
/pas+i/	→	**[pasi]**	/paʃ+i/	→	**[paʃi]**

- Lexically-specified process?
- General process with lexical exceptions?