Generalization despite variation: French schwa with lexically indexed constraints

Aleksei Nazarov
Brian Smith

Overview

1. Over- and underfitting
2. Models of finding lexically specific constraints
3. Case study: French schwa deletion
4. Simulations and results
5. Discussion and wrap-up

Over- and underfitting

Classic problem: child creates grammar that accounts for seen data, generalizes to unseen data (e.g., SPE)

Two potential problems:

Underfitting = not accounting for seen data

Overfitting = not generalizing to unseen data

Especially important for exceptions: account for seen exceptions, generalize to unseen items despite exceptions

Typical tradeoff

If less underfitting: more overfitting
(✓exceptions → ✗generalization)

If less overfitting: more underfitting
(✓generalization → ✗exceptions)

Bias-variance tradeoff;
E.g., Geman et al. (1992),
Hastie et al. (2001)

Models with indexed constraints (Kraska-Szlenk 1995, Pater 2000) or cophonologies (e.g., Inkelas & Zoll 2007):

How strong is this tradeoff?

Are indexed Cs/cophonologies “worth the trouble”?

Our models

Indexed constraint MaxEnt models

Building on existing learners that expand grammar with indexed (lexically-specific) constraints
(Becker 2009, Round 2017, Nazarov 2021)

Grammar framework: MaxEnt (Goldwater & Johnson 2003)

- Can be fit to data with general-purpose learners
- Good at variation (French case study has variation)

Differences between models

1. How are indexed constraints chosen (induced)?

No indexation, Pre-training, Post-training, Iterative

more steps

2. How are indexed constraints generalized to novel words?

0 method, Probabilistic method

more steps

Constraint induction: pre- vs. post-training

Every constraint receives 1 lexically specific variant

Which words are associated w lexically specific constraints :

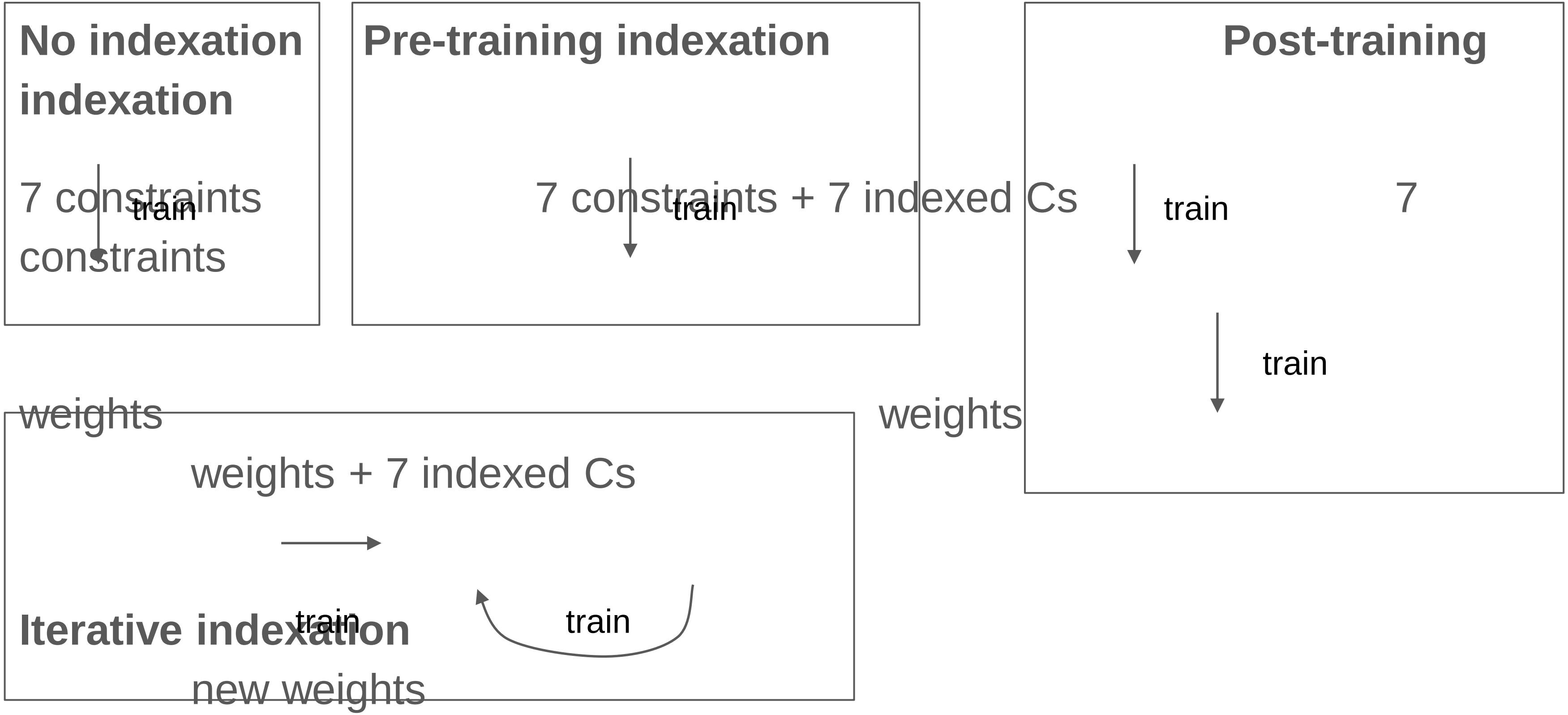
- **Pre-training**: determined based on winner-loser patterns alone, before training the model
- **Post-training**: determined based on estimates of model after one round of training

Constraint induction: iterative

Like post-training induction method, but add one lexically specific (indexed) constraint at a time (cf. Nazarov 2018)

1. Train model without indexed constraints
2. Add highest-impact* indexed constraint
3. Train this updated model again (on the same data)
4. Repeat steps 2-3 until convergence

Constraint induction: summary



No indexation indexation

7 constraints

train

constraints

weights

Pre-training indexation

7 constraints + 7 indexed Cs

train

weights

Post-training

train

7

train

Iterative indexation

weights + 7 indexed Cs

train

train

new weights

Generalization methods

How are properties of exceptions generalized to unseen words?

0 method: unseen words cannot violate lexically specific constraints; after (Pater 2000)

Probabilistic method: unseen words violate lexically specific constraints, scaled to how common the exceptions are in lexicon; after Becker (2009)

Case study: French schwa

French schwa deletion

1. Well-studied phenomenon with relatively well understood phonological conditioning factors
2. Optional phonological process with different degrees of optionality

(never ... almost never .... sometimes … most of the time … always)

Contextually modulated variation

‘Schwa’ [œ] (here: /ə/) variably deleted; depends on context (e.g., Dell 1985)

VC_CV: baseline case; kasəʁɔl ~ kasʁɔl ‘pot’

#C_C: (slightly less deletion); səʁɛ̃ ~ sʁɛ̃ ‘canary’

C_CC/CC_C: much less deletion; subʁəso ~ subʁso ‘jolt’

Exceptions

In addition to contextual influence, also lexical influence, e.g.:

- /sə̲mɛn/ ‘week’ (50% deletion) /sə̲mɛstʁ/ ‘semester’ (14% deletion)

Among words with same context but different deletion rates:

- **Trend-followers**: deletion rate same side of 50% as average across words with this context

- **Exceptions**: deletion rate other side of 50% as average across words with this context

Data

From Racine’s (2008:ch 3) experiment: France French data

456 words with schwa in VC_CV, #C_C, C_CC, or CC_C

    After exclusions based on morphological criteria

Schwa-ful, schwa-less variants of words judged on 1-7 scale (averaged across 12 speakers from Loire-Atlantique region)

Judgments transformed into (pseudo-)frequencies (Appendix)

Constraints used (based on Kaplan 2011)

2-candidate tableaux for each word (e.g., səmɛstʁ vs. smɛstʁ)

*ə to motivate

schwa deletion

*ə[^.σ] no schwa

except in penult σs

Max to motivate

schwa retention

*CCC schwa stays to

avoid CCC clusters

Simulations

Simulation setup

Models: no indexation, pre-training, post-training, iterative

1. To test **underfitting**: train models on entire dataset

   How well are training data predicted by model?

2. To test **overfitting**: train models on various subsets of data (20-fold cross-validation)

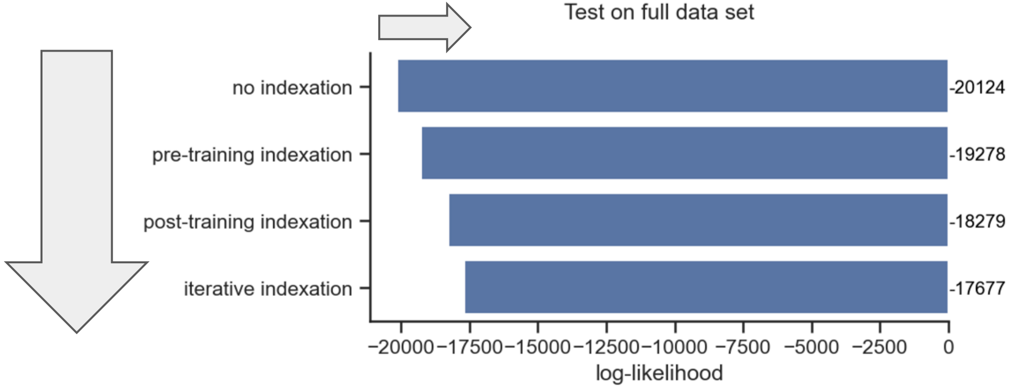
   How well can you predict unseen (held-out) data?

Underfitting test: results

Train each model on entire dataset (456 words)

Test: log-likelihood of entire dataset

(less negative = less underfitting)



(More involved) indexation decreases underfitting

Grammars: Appendix

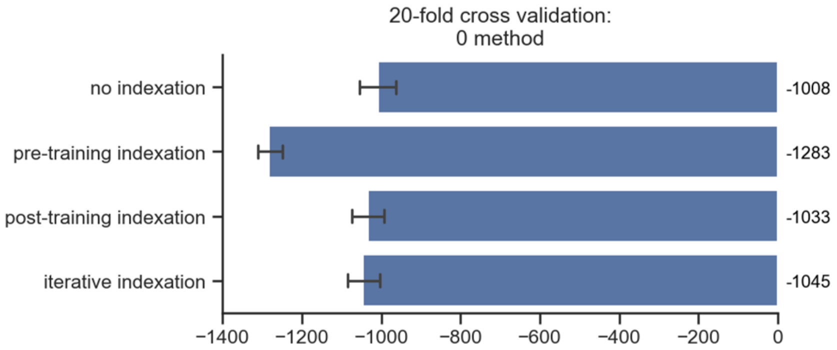
Overfitting test: results

Train each model on 19/20 of data

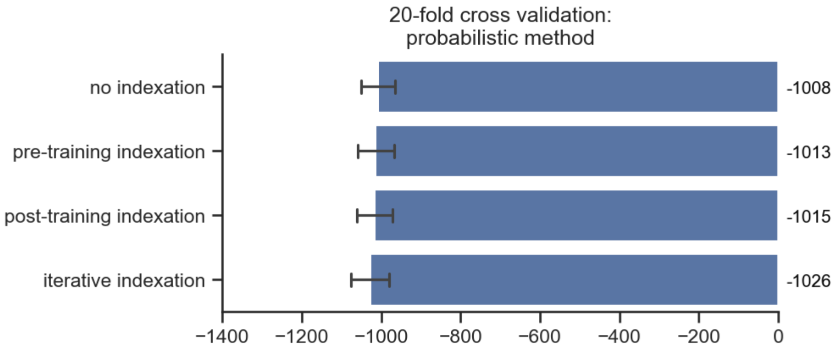
Test: log-likelihood on **remaining** 1/20 of data

(less negative = less overfitting)

Repeat 20 times, leaving out another 1/20 of data each time; then compute mean and 95% CIs



20-fold cross validation:
0 method



20-fold cross validation:
probabilistic method

Indexation does not significantly increase overfitting!

(except pre-training indexation with 0 method generalization)

Discussion/wrap-up

Gradient-based separation & robustness

New: MaxEnt-based induction of lexically specific constraints for exceptional words (generalization of Becker 2009, Pater 2010 for categorical OT)

Can be simple (pre-training) to complicated (iterative)

No matter which one you use, you will better model patterns & exceptions, but not significantly impact generalization

(*decrease underfitting without increasing overfitting*)

Role of complexity

More sophisticated models do better on exceptions, but even simplest indexation model helps (*decreases underfitting*)

Iterative indexation: fewer constraints, but less underfitting!

However, simplest model + 0 indexation doesn’t work!

Indexation doesn’t take constraint interaction into account

Majority of trend-followers associated with lexically specific

Future work

Apply to datasets with more constraints, more candidates

- Will this change relative advantage of sophisticated models? Will properties of simplest indexation model remain?

Further investigation of iterative indexation model

- How conservative is it? Lexicon-grammar divide?
- Compare to older work of this kind (e.g. Nazarov 2018)

Thank you!

References

- Becker, M. 2009. *Phonological Trends in the Lexicon: The Role of Constraints.* University of Massachusetts Amherst dissertation.
- Byrd, R.H., P. Lu & J. Nocedal. 1995. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific and Statistical Computing*, 16.5:1190-1208.
- Chomsky N. & M. Halle. 1968. *The Sound Pattern of English*. New York: Harper & Row.
- Dell, F. 1985. *Les règles et les sons: Introduction à la phonologie générative*. Paris: Hermann.
- Geman, S., E. Bienenstock & R. Doursat. 1992. Neural networks and the bias/variance dilemma. *Neural Computation* 4.1:1-58.
- Goldwater, S. & M. Johnson. 2003. Learning OT Constraint Rankings Using a Maximum Entropy Model. In J. Spenader, A. Eriksson, Ö. Dahl (eds.), Proceedings of the Stockholm Workshop on Variation within Optimality Theory, 111–20.
- Hastie, T., R. Tibshirani, R. & J. Friedman. 2001. *The Elements of Statistical Learning*. New York: Springer.
- Hayes, B. & Z. Londe. 2006. Stochastic Phonological Knowledge: The Case of Hungarian Vowel Harmony. *Phonology* 23:59-104.
- Inkelas, S. & C. Zoll. 2007. Is Grammar Dependence Real? A Comparison Between Cophonological and Indexed Constraint Approaches to Morphologically Conditioned Phonology. *Linguistics* 45.1:133–71.

References

- Kaplan, A. 2011. Variation through Markedness Suppression. Phonology 28.3:331–370.
- Kraska-Szlenk, I. 1995. *The Phonology of Stress in Polish.* University of Illinois at Urbana-Champaign dissertation.
- Nazarov, A. 2018. ‘Learning within- and between-word variation in probabilistic OT grammars.’ In G. Gallagher, M. Gouskova, and S. Yin (eds.), *Supplemental Proceedings of the 2017 Annual Meeting on Phonology*, Washington, DC: LSA.
- Pater, J. 2000. Non-uniformity in English secondary stress: The role of ranked and lexically specific constraints. *Phonology* 17.2:237–74.
- Racine, I. 2008. Les effets de l'effacement du Schwa sur la production et la perception de la parole en français. Université de Genève dissertation.
- Round, E. 2017. Phonological exceptionality is localized to phonological elements: The argument from learnability and Yidiny word-final deletion. In C. Bowern, L. Horn & R. Zanuttini (eds.), On looking into words (and beyond): Structures, relations, analyses, 59–97. Berlin: Language Science Press.
- Staubs, R. 2011. *Harmonic Grammar in R (hgR).* Software package. http://blogs.umass.edu/hgr/

Appendix

Pseudo-frequencies

Schwa-ful, schwa-less variants of words judged on 1-7 scale (averaged across 12 speakers from Loire-Atlantique region)

> Make into (pseudo-)frequencies: subtract 1 from all judgments (0-6 range), then divide the each judgment by the sum of judgments for that word (proportion)

E.g.

$$\frac{J(\text{smɛstʁ}) - 1}{J(\text{smɛstʁ}) - 1 + J(\text{səmɛstʁ}) - 1} = 0.92/(0.92+5.50) = 0.14$$

How is indexation learned?

For each word and each constraint:

compute the **gradient** (derivative) of the constraint’s weight (= how much does this word prefer for the weight to go up or down?)

When constraint is given an indexed version:

associate indexed version exclusively with words that yield a **positive gradient** (that want a higher ranking for this constraint)

What is the highest-impact indexed constraint?

For each potential indexed constraint compute Mean Absolute Error (MAE) of the gradients:

- deviations of individual words’ gradients from the mean gradient: <-.05, +.02, +.03>
- absolute of these gradients: <.05, .02, .03>
- mean of these absolute gradients: .033

Highest-impact indexed constraint = constraint with max MAE

Simulation setup

Models: no indexation, pre-training, post-training, iterative

Trained with L-BFGS-B method (Byrd et al. 1995), using Staubs' (2011) implementation; L2 prior: μ=0, σ^2=1,000,000

1. To test underfitting: train models on entire dataset

   How well are training data predicted by model?

2. To test overfitting: 20-fold cross-validation

   How well can you predict unseen (held-out) data?

Train on entire dataset: resulting grammars

No indexation			Pre-training indexation			Post-training indexation			Iterative indexation		
Constr	Weight	% of R.wds	Constr	Weight	% of R.wds	Constr	Weight	% of R.wds	Constr	Weight	% of R.wds
*CNC	1.56	100%	Max$_i$	1.25	87%	*CNC	1.80	100%	*CNC	1.88	100%
Max	1.14	100%	*CNC	0.79	100%	Max$_i$	1.20	54%	Max$_i$	1.83	54%
*CCC	0.92	100%	*CNC$_j$	0.79	100%	Max	1.11	100%	Max	1.22	100%
*ə[^.σ]	0.29	100%	*ə$_k$	0.65	1%	*CTN	0.78	100%	*CCC	0.89	100%
*CTN	0.26	100%	Max	0.47	100%	*CCC	0.61	100%	*ə$_j$	0.78	46%
*ə	0.004	100%	*CCC	0.289	100%	*CCC$_j$	0.46	60%	*CTN	0.76	100%
*#CC	0.00	100%	*CCC$_m$	0.289	97%	*ə$_k$	0.30	46%	*ə	0.56	100%
			*ə[^.σ]	0.287	100%	*ə	0.30	100%	*#CC$_k$	0.48	47%
			*ə	0.22	100%	*ə[^.σ]	0.23	100%	*ə$_{j,m}$	0.32	22%
			*CTN	0.15	100%	*CNC$_m$	0.06	57%	*ə[^.σ]	0.31	100%
			*CTN$_n$	0.15	100%	*#CC	0.00	100%	*#CC	0.00	100%
			*#CC	0.00	100%	*CTN$_n$	0.00	40%			
			*#CC$_p$	0.00	73%	*#CC$_p$	0.00	44%			
			*ə[^.σ]$_q$	0.00	1%	*ə[^.σ]$_q$	0.00	53%			

Parts of pattern missed: *ə has practically no weight

Indexed constraints apply to (almost) all or (almost) no relevant inputs

Some indexed Cs' weights close to non-indexed Cs

Doubly-indexed constraint: layers of exceptionality

Example tableau: no indexation

/səmɛn/ ‘week’ (50% deletion) /səmɛstʁ/ ‘semester’ (14% deletion)

				0.92	0.004	0	1.14
input	output	observed probability	predicted probability	*CCC	*ə	*#CC	Max
/səmɛn/	səmɛn	50%	76%	0	-1	0	0
	smɛn	50%	24%	0	0	-1	-1
/səmɛstʁ/	səmɛstʁ	86%	76%	-1	-1	0	0
	smɛstʁ	14%	24%	-1	0	-1	-1

Example tableau: pre-training indexation

/sə̲mɛn/ ‘week’ (50% deletion)

/sə̲mɛstʁ/ ‘semester’ (14% deletion)

				0.29	0.22	0	0.47	0	1.25
input	output	observed probability	predicted probability	*CCC	*ə	*#CC	Max	*#CC$_p$	*Max$_i$
/səmɛn/	səmɛn	50%	56%	0	-1	0	0	0	0
	smɛn	50%	44%	0	0	-1	-1	0	0
/səmɛstʁ/	səmɛstʁ	86%	82%	-1	-1	0	0	0	0
	smɛstʁ	14%	18%	-1	0	-1	-1	-1	-1

Example tableau: post-training indexation

/sə̲mɛn/ ‘week’ (50% deletion) /sə̲mɛstʁ/ ‘semester’ (14% deletion)

				0.61	0.30	0	1.11	0.30	0	1.20
input	output	observed probability	predicted probability	*CCC	*ə	*#CC	Max	*ə$_k$	*#CC$_p$	*Max$_i$
/səmɛn/	səmɛn	50%	62%	0	-1	0	0	-1	0	0
	smɛn	50%	38%	0	0	-1	-1	0	0	0
/səmɛstʁ/	səmɛstʁ	86%	88%	-1	-1	0	0	0	0	0
	smɛstʁ	14%	12%	-1	0	-1	-1	0	-1	-1

Example tableau: iterative indexation

/sə̲mɛn/ ‘week’ (50% deletion) /sə̲mɛstʁ/ ‘semester’ (14% deletion)

				0.89	0.56	0	1.22	0.78	0.32	0.48	1.83
input	output	observed probability	predicted probability	*CCC	*ə	*#CC	Max	*ə$_{j}$	*ə$_{j,m}$	*#CC$_{k}$	Max$_{i}$
/səmɛn/	səmɛn	50%	47%	0	-1	0	0	-1	0	0	0
	smɛn	50%	53%	0	0	-1	-1	0	0	0	0
/səmɛstʁ/	səmɛstʁ	86%	87%	-1	-1	0	0	-1	-1	0	0
	smɛstʁ	14%	13%	-1	0	-1	-1	0	0	-1	-1