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 → X generalization)

If less overfitting: more underfitting
(✔ generalization → X 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

Pre-training indexation
7 constraints + 7 indexed Cs

Post-training
7

Iterative indexation
weights + 7 indexed Cs

train

weights

train

train
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 ~ subʁ̥so ‘jolt’
Exceptions

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

/ˈwɛk/ ‘week’ (50% deletion)   /ˈsemɛstər/ ‘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., əmɛstʁ vs. smɛstʁ)

*ə
to motivate schwa deletion

*ə[^.σ]
no schwa except in penult əσs

Max
to motivate schwa retention

*CCC
schwa stays to avoid CCC cluster
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)

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 constraints lead to overfitting
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

References

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. \[ \frac{J(\text{smɛstʁ}) - 1}{J(\text{smɛstʁ}) - 1 + J(\text{səmɛstʁ}) - 1} = \frac{0.92}{(0.92+5.50)} = 0.14 \]
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: $\mu=0$, $\sigma^2=1,000,000$

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

<table>
<thead>
<tr>
<th>No indexation</th>
<th>Pre-training indexation</th>
<th>Post-training indexation</th>
<th>Iterative indexation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constr</td>
<td>Weight</td>
<td>% of R.wds</td>
<td>Constr</td>
</tr>
<tr>
<td>*CNC</td>
<td>1.56</td>
<td>100%</td>
<td>Max_i</td>
</tr>
<tr>
<td>Max</td>
<td>1.14</td>
<td>100%</td>
<td>*CNC</td>
</tr>
<tr>
<td>*CCC</td>
<td>0.92</td>
<td>100%</td>
<td>*CNC_j</td>
</tr>
<tr>
<td>*ə[^.σ]</td>
<td>0.29</td>
<td>100%</td>
<td>*ə_k</td>
</tr>
<tr>
<td>*CTN</td>
<td>0.26</td>
<td>100%</td>
<td>Max</td>
</tr>
<tr>
<td>*ə</td>
<td>0.004</td>
<td>100%</td>
<td>*CCC</td>
</tr>
<tr>
<td>*#CC</td>
<td>0.00</td>
<td>100%</td>
<td>*CCC_m</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>*ə[^.σ]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>*ə</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>*CTN</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>*CTN_7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>*#CC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>*#CC_p</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>*ə[^.σ]_q</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>/səmɛn/ ‘week’ (50% deletion)</th>
<th>/səmɛstəɾ/ ‘semester’ (14% deletion)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
<td>output</td>
<td>observed probability</td>
<td>predicted probability</td>
<td>*CCC</td>
<td>*ə</td>
</tr>
<tr>
<td>/səmɛn/</td>
<td>səmɛn</td>
<td>50%</td>
<td>76%</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>smɛn</td>
<td>50%</td>
<td>24%</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>/səmɛstəɾ/</td>
<td>səmɛstəɾ</td>
<td>86%</td>
<td>76%</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>smɛstəɾ</td>
<td>14%</td>
<td>24%</td>
<td>-1</td>
<td>0</td>
</tr>
</tbody>
</table>
### Example tableau: pre-training indexation

<table>
<thead>
<tr>
<th>/səmɛn/ ‘week’ (50% deletion)</th>
<th>/səmɛstɾ/ ‘semester’ (14% deletion)</th>
<th>Input</th>
<th>Output</th>
<th>Observed Probability</th>
<th>Predicted Probability</th>
<th>CCC</th>
<th>ø</th>
<th>#CC</th>
<th>Max</th>
<th>#CCp</th>
<th>Max₀</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>input</strong></td>
<td><strong>output</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/səmɛn/</td>
<td>səmɛn</td>
<td>50%</td>
<td>56%</td>
<td></td>
<td></td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>smɛn</td>
<td>50%</td>
<td>44%</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>/səmɛstɾ/</td>
<td>səmɛstɾ</td>
<td>86%</td>
<td>82%</td>
<td></td>
<td></td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>smɛstɾ</td>
<td>14%</td>
<td>18%</td>
<td></td>
<td></td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>
Example tableau: post-training indexation

<table>
<thead>
<tr>
<th>/səmɛn/ ‘week’ (50% deletion)</th>
<th>/səmɛstɾ/ ‘semester’ (14% deletion)</th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
<td>output</td>
</tr>
<tr>
<td>/səmɛn/</td>
<td>səmɛn</td>
</tr>
<tr>
<td></td>
<td>smɛn</td>
</tr>
<tr>
<td>/səmɛstɾ/</td>
<td>səmɛstɾ</td>
</tr>
<tr>
<td></td>
<td>smɛstɾ</td>
</tr>
</tbody>
</table>
## Example tableau: iterative indexation

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
<th>Observed Probability</th>
<th>Predicted Probability</th>
<th>CCC</th>
<th>*ə</th>
<th>#CC</th>
<th>Max</th>
<th>*ə&lt;sub&gt;j&lt;/sub&gt;</th>
<th>*ə&lt;sub&gt;j,m&lt;/sub&gt;</th>
<th>#CC&lt;sub&gt;k&lt;/sub&gt;</th>
<th>Max&lt;sub&gt;i&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>/səmɛn/</td>
<td>/səmɛn/</td>
<td>50%</td>
<td>47%</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>smɛn</td>
<td>50%</td>
<td>53%</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>/səmɛstɾ/</td>
<td>/səmɛstɾ/</td>
<td>86%</td>
<td>87%</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>smɛstɾ</td>
<td>14%</td>
<td>13%</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
</tbody>
</table>