Errors in language production, language learning and language change
Some computational experiments with Optimality Theory

Tamás Biró
ACLC, University of Amsterdam (UvA)

OAP dag, December 16, 2011
Example: sentential negation (Jespersen’s cycle)

<table>
<thead>
<tr>
<th></th>
<th>pre-verbal</th>
<th>discontinuous</th>
<th>post-verbal</th>
</tr>
</thead>
<tbody>
<tr>
<td>French</td>
<td>Je ne dis</td>
<td>Je ne dis pas</td>
<td>Je dis pas</td>
</tr>
<tr>
<td>English</td>
<td>Ic ne secge</td>
<td>Ic ne seye not</td>
<td>I say not</td>
</tr>
</tbody>
</table>

1. **SN V**  
2. **SN V SN**  
3. **V SN**

To explain:
- Typology: pre-verbal, discontinuous, post-verbal,
- ... as well as mixed types.
- Diachronic change (a.k.a. language evolution).
Example: sentential negation (Jespersen’s cycle)

<table>
<thead>
<tr>
<th></th>
<th>pre-verbal</th>
<th>discontinuous</th>
<th>post-verbal</th>
</tr>
</thead>
<tbody>
<tr>
<td>French</td>
<td>Je ne dis</td>
<td>Je ne dis pas</td>
<td>Je dis pas</td>
</tr>
<tr>
<td>English</td>
<td>Ic ne sece</td>
<td>Ic ne seye not</td>
<td>I say not</td>
</tr>
<tr>
<td>1.</td>
<td>SN V</td>
<td>2. SN V SN</td>
<td>3. V SN</td>
</tr>
</tbody>
</table>

To explain:

- **Typology:** pre-verbal, discontinuous, post-verbal,
- ... as well as mixed types.
- **Diachronic change** (a.k.a. language evolution).
Example: sentential negation (Jespersen’s cycle)

<table>
<thead>
<tr>
<th></th>
<th>pre-verbal</th>
<th>discontinuous</th>
<th>post-verbal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>French</strong></td>
<td>Je ne dis</td>
<td>Je ne dis pas</td>
<td>Je dis pas</td>
</tr>
<tr>
<td><strong>English</strong></td>
<td>Ic ne sece</td>
<td>Ic ne seye not</td>
<td>I say not</td>
</tr>
<tr>
<td></td>
<td>1. $SN \ V$</td>
<td>2. $SN \ V \ SN$</td>
<td>3. $V \ SN$</td>
</tr>
</tbody>
</table>

To explain:
- Typology: pre-verbal, discontinuous, post-verbal,
- ... as well as mixed types.
- Diachronic change (a.k.a. language evolution).
Example: sentential negation (Jespersen’s cycle)

<table>
<thead>
<tr>
<th></th>
<th>pre-verbal</th>
<th>discontinuous</th>
<th>post-verbal</th>
</tr>
</thead>
<tbody>
<tr>
<td>French</td>
<td>Je ne dis</td>
<td>Je ne dis pas</td>
<td>Je dis pas</td>
</tr>
<tr>
<td>English</td>
<td>Ic ne seyce</td>
<td>Ic ne seye not</td>
<td>I say not</td>
</tr>
<tr>
<td></td>
<td>1. $SN : V$</td>
<td>2. $SN : V : SN$</td>
<td>3. $V : SN$</td>
</tr>
</tbody>
</table>

To explain:
- Typology: pre-verbal, discontinuous, post-verbal,
- ... as well as mixed types.
- Diachronic change (a.k.a. language evolution).
Research questions:

The role of *errors* = the results of imperfect mental computation.

- “Performance errors”: ungrammatical but produced.
- Learning in the presence of “performance errors”.
- “Performance errors” as a driving force behind language change.
- Another reason for making errors during learning.

References

- T. Biró. ‘Towards a Robuster Interpretive Parsing: Learning from overt forms in Optimality Theory’. Submitted to *Journal of Logic, Language and Information*.
Research questions:

The role of *errors* = the results of imperfect mental computation.

- “Performance errors”: ungrammatical but produced.
- Learning in the presence of “performance errors”.
- “Performance errors” as a driving force behind language change.
- Another reason for making errors during learning.

References

- T. Biró. ‘Towards a Robuster Interpretive Parsing: Learning from overt forms in Optimality Theory’. Submitted to *Journal of Logic, Language and Information*. 

Tamás Biró

Errors in language production, language learning and language change

3/30
Research questions:

The role of *errors* = the results of imperfect mental computation.

- “Performance errors”: ungrammatical but produced.
- Learning in the presence of “performance errors”.
- “Performance errors” as a driving force behind language change.
- Another reason for making errors during learning.

References

- T. Biró. ‘Towards a Robuster Interpretive Parsing: Learning from overt forms in Optimality Theory’. Submitted to *Journal of Logic, Language and Information*.
Research questions:

The role of \textit{errors} = the results of imperfect mental computation.

- “Performance errors”: ungrammatical but produced.
- Learning in the presence of “performance errors”.
- “Performance errors” as a driving force behind language change.
- Another reason for making errors during learning.

References

Errors of the mental computation

static knowledge processes in the brain

competence

performance
Errors of the mental computation

static knowledge processes in the brain
The language acquisition problem
Learning from competence?
Learning from performance!

COMPETENCE \[\rightarrow\] PERFORMANCE

LEARNING \[\leftarrow\] PERFORMANCE

Tamás Biró
Errors in language production, language learning and language change
7/30
## Overview

1. Modelling linguistic performance
2. Issues in learning and iterated learning
3. The problem of the overt forms
4. Conclusions
Overview

1. Modelling linguistic performance
2. Issues in learning and iterated learning
3. The problem of the overt forms
4. Conclusions
Errors of the mental computation

static knowledge
Optimality Theory

processes in the brain
Simulated Annealing for OT
Errors of the mental computation

static knowledge
Optimality Theory

processes in the brain
Simulated Annealing for OT
Errors of the mental computation

A grammar is a Harmony function on the candidate set, defined by the ranked constraints.
Global optimum: more harmonic than all other candidates.
Local optimum: more harmonic than its neighbours.

<table>
<thead>
<tr>
<th>/aat/</th>
<th>NoCoda</th>
<th>Parse</th>
<th>Onset</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.a&lt;t&gt;</td>
<td>*</td>
<td>*</td>
<td>**</td>
</tr>
<tr>
<td>a.at</td>
<td>*!</td>
<td></td>
<td>**</td>
</tr>
<tr>
<td>&lt;a&gt;at</td>
<td>*!</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

**Optimality Theory**

Grammar competence model

grammatical form = $\mathcal{F}$ (globally) optimal candidate

**SA-OT**

Implementation performance model

produced forms = globally or locally optimal candidates
Modelling linguistic competence

\[ \text{Faith[Neg]} \gg *\text{Negation} \gg \text{NegationFirst} \gg \text{NegationLast} \]

<table>
<thead>
<tr>
<th>/pol = neg/</th>
<th>Faith[Neg]</th>
<th>*Neg</th>
<th>NegFirst</th>
<th>NegLast</th>
</tr>
</thead>
<tbody>
<tr>
<td>([V])</td>
<td>*</td>
<td></td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>([SN \ V])</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>([V \ SN])</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>([SN \ V \ SN])</td>
<td></td>
<td></td>
<td>**</td>
<td></td>
</tr>
<tr>
<td>([V \ SN \ SN])</td>
<td></td>
<td></td>
<td>**</td>
<td>*</td>
</tr>
<tr>
<td>([SN \ SN \ V])</td>
<td></td>
<td></td>
<td>**</td>
<td>*</td>
</tr>
<tr>
<td>([SN \ V \ SN \ SN])</td>
<td></td>
<td></td>
<td>***</td>
<td></td>
</tr>
<tr>
<td>[\ldots]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Modelling linguistic competence

**Faith[Neg] >> NegationFirst >> *Negation >> NegationLast**

<table>
<thead>
<tr>
<th>/pol = neg/</th>
<th>Faith[Neg]</th>
<th>NegFirst</th>
<th>*Neg</th>
<th>NegLast</th>
</tr>
</thead>
<tbody>
<tr>
<td>[V]</td>
<td>*</td>
<td>*</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>[SN V]</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>[V SN]</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[SN V SN]</td>
<td></td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>[V SN SN]</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[SN SN V]</td>
<td></td>
<td></td>
<td></td>
<td>**</td>
</tr>
<tr>
<td>[SN V SN SN]</td>
<td></td>
<td></td>
<td></td>
<td>***</td>
</tr>
</tbody>
</table>

...  

Errors of the mental computation

A grammar is a Harmony function on the candidate set, defined by the ranked constraints. Global optimum: more harmonic than all other candidates. Local optimum: more harmonic than its neighbours.

**Optimality Theory**

Grammar competence model

Grammatical form = \( \mathfrak{F} \) (globally) optimal candidate

Produced forms = globally or locally optimal candidates

**SA-OT**

Implementation performance model
Modelling linguistic performance

A topology (neighborhood structure) on the candidate set:

Locally optimal forms: are predicted to be the produced forms.
Modelling linguistic performance

Faith[Neg] ≫ *Negation ≫ NegationFirst ≫ NegationLast

Locally optimal forms: ⚛ [SN V].
Modelling linguistic performance

\[ \text{FAITH[NEG]} \gg \text{NEGATIONFIRST} \gg \ast \text{NEGATION} \gg \text{NEGATIONLAST} \]

Locally optimal forms: \(\bowtie [\text{SN V}]\) and \(\sim [\text{SN [V SN]}]\).
<table>
<thead>
<tr>
<th>Hierarchy</th>
<th>competence</th>
<th>performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. *Neg ≫ NegFirst ≫ NegLast</td>
<td>pre-verbal</td>
<td>pre-verbal</td>
</tr>
<tr>
<td>2. NegFirst ≫ *Neg ≫ NegLast</td>
<td>pre-verbal</td>
<td>pre-V and discont.</td>
</tr>
<tr>
<td>3. NegFirst ≫ NegLast ≫ *Neg</td>
<td>discontinuous</td>
<td>discontinuous</td>
</tr>
<tr>
<td>4. NegLast ≫ NegFirst ≫ *Neg</td>
<td>discontinuous</td>
<td>discontinuous</td>
</tr>
<tr>
<td>5. NegLast ≫ *Neg ≫ NegFirst</td>
<td>post-verbal</td>
<td>discont. and post-V</td>
</tr>
<tr>
<td>6. *Neg ≫ NegLast ≫ NegFirst</td>
<td>post-verbal</td>
<td>post-verbal</td>
</tr>
</tbody>
</table>

**Observerd typology:** 3 pure types and 2 mixed types.

**Predicted typology:**

- Traditional OT (H. de Swart): 3 pure types.
- Stochastic OT (H. de Swart): 3 pure types and 3 mixed types.
- SA-OT (Lopopolo and Biró): 3 pure types and 2 mixed types.
Modelling performance

Learning

Overt forms

Conclusions

Table:

<table>
<thead>
<tr>
<th>Hierarchy</th>
<th>competence</th>
<th>performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $*\text{Neg} \gg \text{NegFirst} \gg \text{NegLast}$</td>
<td>pre-verbal</td>
<td>pre-verbal</td>
</tr>
<tr>
<td>2. $\text{NegFirst} \gg *\text{Neg} \gg \text{NegLast}$</td>
<td>pre-verbal</td>
<td>pre-V and discont.</td>
</tr>
<tr>
<td>3. $\text{NegFirst} \gg \text{NegLast} \gg *\text{Neg}$</td>
<td>discontinuous</td>
<td>discontinuous</td>
</tr>
<tr>
<td>4. $\text{NegLast} \gg \text{NegFirst} \gg *\text{Neg}$</td>
<td>discontinuous</td>
<td>discontinuous</td>
</tr>
<tr>
<td>5. $\text{NegLast} \gg *\text{Neg} \gg \text{NegFirst}$</td>
<td>post-verbal</td>
<td>discont. and post-V</td>
</tr>
<tr>
<td>6. $*\text{Neg} \gg \text{NegLast} \gg \text{NegFirst}$</td>
<td>post-verbal</td>
<td>post-verbal</td>
</tr>
</tbody>
</table>

**Observerd typology:** 3 pure types and 2 mixed types.

**Predicted typology:**

- Traditional OT (H. de Swart): 3 pure types.
- Stochastic OT (H. de Swart): 3 pure types and 3 mixed types.
- SA-OT (Lopopolo and Biró): 3 pure types and 2 mixed types.
<table>
<thead>
<tr>
<th>Hierarchy</th>
<th>competence</th>
<th>performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. *Neg ≫ NegFirst ≫ NegLast</td>
<td>pre-verbal</td>
<td>pre-verbal</td>
</tr>
<tr>
<td>2. NegFirst ≫ *Neg ≫ NegLast</td>
<td>pre-verbal</td>
<td>pre-V and discont.</td>
</tr>
<tr>
<td>3. NegFirst ≫ NegLast ≫ *Neg</td>
<td>discontinuous</td>
<td>discontinuous</td>
</tr>
<tr>
<td>4. NegLast ≫ NegFirst ≫ *Neg</td>
<td>discontinuous</td>
<td>discontinuous</td>
</tr>
<tr>
<td>5. NegLast ≫ *Neg ≫ NegFirst</td>
<td>post-verbal</td>
<td>discont. and post-V</td>
</tr>
<tr>
<td>6. *Neg ≫ NegLast ≫ NegFirst</td>
<td>post-verbal</td>
<td>post-verbal</td>
</tr>
</tbody>
</table>

**Observerd typology:** 3 pure types and 2 mixed types.

**Predicted typology:**
- Traditional OT (H. de Swart): 3 pure types.
- Stochastic OT (H. de Swart): 3 pure types and 3 mixed types.
- SA-OT (Lopopolo and Biró): 3 pure types and 2 mixed types.
Overview

1. Modelling linguistic performance
2. Issues in learning and iterated learning
3. The problem of the overt forms
4. Conclusions
Iterated learning: reproducing language change (?)

Five agents in each generation. Generations 0 to 100. Each agent learns from every agent in the previous generation. Negation types in the “simulated historical corpus”:


Learning from performance!
Language acquisition with online learning algorithms

Δ = ?

Δ = ?

PERFORMANCE

LEARNING

COMPETENCE

COMPETENCE

PERFORMANCE
Online learning algorithms

Constraint $C_i$ has rank $r_i$.

In each learning cycle: learning data (winner) produced by teacher compared to form produced by learner (loser).

Update rule: update the rank $r_i$ of every constraint $C_i$, depending on whether $C_i$ prefers the winner or the loser.

- Run until convergence of performance, and not of competence.
- Distance of teacher sample vs. learner sample measured by JSD:

Jensen-Shannon divergence: measures the “distance” of two distributions

$$JSD(P\|Q) = \frac{D(P\|M) + D(Q\|M)}{2}$$

where $D(P\|Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$ (relative entropy, Kullback-Leibler divergence), $M(x) = \frac{P(x)+Q(x)}{2}$. 
Online learning algorithms

Constraint $C_i$ has rank $r_i$.

In each learning cycle: learning data (winner) produced by teacher compared to form produced by learner (loser).

Update rule: update the rank $r_i$ of every constraint $C_i$, depending on whether $C_i$ prefers the winner or the loser.

- Run until convergence of performance, and not of competence.
- Distance of teacher sample vs. learner sample measured by JSD:

Jensen-Shannon divergence: measures the “distance” of two distributions

$$JSD(P \parallel Q) = \frac{D(P \parallel M) + D(Q \parallel M)}{2}$$

where $D(P \parallel Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$ (relative entropy, Kullback-Leibler divergence), $M(x) = \frac{P(x) + Q(x)}{2}$. 
Results: number of learning steps until convergence

- Measure the number of learning steps until convergence.
- 2000 times learning (rnd target, rnd underlying form) per grammar type \( \times \) production method \( \times \) learning method.
- Long-tail distribution of number of learning steps:

<table>
<thead>
<tr>
<th>production</th>
<th>update rule</th>
<th>OT</th>
<th>10-HG</th>
<th>4-HG</th>
</tr>
</thead>
<tbody>
<tr>
<td>grammatical</td>
<td>Magri</td>
<td>13 ; 27 ; 45 ; 67</td>
<td>13 ; 28 ; 46 ; 70</td>
<td>12 ; 27 ; 48 ; 69</td>
</tr>
<tr>
<td></td>
<td>GLA</td>
<td>23 ; 43 ; 65 ; 102</td>
<td>22 ; 41 ; 64 ; 107</td>
<td>22 ; 42 ; 64 ; 107</td>
</tr>
<tr>
<td>SA-OT, ( t_{step} = 0.1 )</td>
<td>Magri</td>
<td>53 ; 109 ; 233 ; 497</td>
<td>63 ; 140 ; 328 ; 1681</td>
<td>60 ; 148 ; 366 ; 1517</td>
</tr>
<tr>
<td></td>
<td>GLA</td>
<td>80 ; 171 ; 462 ; 1543</td>
<td>92 ; 240 ; 772 ; 7512</td>
<td>92 ; 239 ; 785 ; 8633</td>
</tr>
<tr>
<td>SA-OT, ( t_{step} = 1 )</td>
<td>Magri</td>
<td>64 ; 131 ; 305 ; 1022</td>
<td>62 ; 134 ; 304 ; 1127</td>
<td>63 ; 137 ; 329 ; 1278</td>
</tr>
<tr>
<td></td>
<td>GLA</td>
<td>90 ; 212 ; 560 ; 1966</td>
<td>92 ; 233 ; 572 ; 3116</td>
<td>84 ; 212 ; 646 ; 3005</td>
</tr>
</tbody>
</table>

( 1st quartile ; median ; 3rd quartile ; 90th percentile )
Overview

1. Modelling linguistic performance
2. Issues in learning and iterated learning
3. The problem of the overt forms
4. Conclusions
The problem of the overt forms

- Generation $n$ produces $[\text{SN} \ [\text{V SN}]]$ and utters “SN V SN”.
- Generation $n + 1$ hears “SN V SN”.
  Is it $[\text{SN} \ [\text{V SN}]]$ or $[[\text{SN V}] \ \text{SN}]$?

- In general, huge amount of crucial information for the reconstruction of a grammar is covert.
  - Co-indexation: $He_i$ looks like him$_{i/j}$.
  - Foot structure: $\text{banána}$ proof for $ba[nána]$ or $[baná]na$?
  - Basic word order: $\text{John loves Mary}$ proof for SVO or OVS?
- Does it mislead learning?
The problem of the overt forms

- Generation $n$ produces [SN [V SN]] and utters “SN V SN”.
- Generation $n + 1$ hears “SN V SN”.
  Is it [SN [V SN]] or [[SN V] SN]?

- In general, huge amount of crucial information for the reconstruction of a grammar is covert.
  - Co-indexation: $He_i$ looks like $him_{i/j}$.
  - Foot structure: banána proof for ba[nána] or [baná]na?
  - Basic word order: John loves Mary proof for SVO or OVS?

- Does it mislead learning?
The problem of the overt forms

<table>
<thead>
<tr>
<th></th>
<th>*Neg</th>
<th>V-right</th>
<th>V-left</th>
</tr>
</thead>
<tbody>
<tr>
<td>L  $\leftarrow$</td>
<td>[SN V]</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>L's target</td>
<td>[[SN V] SN]</td>
<td>**</td>
<td>*</td>
</tr>
<tr>
<td>T  $\rightarrow$</td>
<td>[SN [V SN]]</td>
<td>**</td>
<td>*</td>
</tr>
</tbody>
</table>

- Learner: *Neg $\gg$ V-right $\gg$ V-left. Produces [SN V].
- Teacher: V-left $\gg$ V-right $\gg$ *Neg. Produces [SN [V SN]].
- Learner hears “SN V SN”. Would like to change her grammar to produce ... [[SN V] SN] or [SN [V SN]]?
- Form [[SN V] SN] is still better than [SN [V SN]] in her grammar, so she takes it as the target for learning,
- ... and fails to learn the target language.
The problem of the overt forms

<table>
<thead>
<tr>
<th>Learner</th>
<th>Teacher</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Neg</td>
<td>V-right</td>
</tr>
<tr>
<td>V-left</td>
<td></td>
</tr>
</tbody>
</table>

Learner: *Neg ≫ V-right ≫ V-left. Produces [SN V].

Teacher: V-left ≫ V-right ≫ *Neg. Produces [SN [V SN]].

Learner hears “SN V SN”. Would like to change her grammar to produce ... [[SN V] SN] or [SN [V SN]]?

Form [[SN V] SN] is still better than [SN [V SN]] in her grammar, so she takes it as the target for learning,

... and fails to learn the target language.
The problem of the overt forms

A (partial) solution:

- Learner hears “SN V SN”. Is it \([SN V] SN\) or \([SN [V SN]]\)?
- Since the learner really cannot know, she takes the (weighted) average of the violations by these forms,
- Teacher produces \([SN [V SN]]\). Learner produces \([SN V]\).

and updates the grammar in order to approach this average.

Learner $\rightarrow$ Teacher

<table>
<thead>
<tr>
<th></th>
<th>*Neg</th>
<th>V-right</th>
<th>V-left</th>
</tr>
</thead>
<tbody>
<tr>
<td>L $\leftarrow$</td>
<td>[SN V]</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>[[SN V] SN]</td>
<td>**</td>
<td>*</td>
</tr>
<tr>
<td>T $\rightarrow$</td>
<td>[SN [V SN]]</td>
<td>**</td>
<td>*</td>
</tr>
<tr>
<td>L’s target</td>
<td>“average”</td>
<td>2</td>
<td>0.5</td>
</tr>
</tbody>
</table>

The improved learning algorithm performs significantly better:
Biró. ‘Towards a Robuster Interpretive Parsing: Learning from overt forms in Optimality Theory’. Submitted to Journal of Logic, Language and Information.
The problem of the overt forms

A (partial) solution:

- Learner hears “SN V SN”. Is it [[SN V] SN] or [SN [V SN]]?
- Since the learner really cannot know, she takes the (weighted) average of the violations by these forms,
- Teacher produces [SN [V SN]]. Learner produces [SN V]. and updates the grammar in order to approach this average.

<table>
<thead>
<tr>
<th></th>
<th>Learner →</th>
<th>Teacher ←</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Neg</td>
<td>V-right</td>
<td>V-left</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>[SN V]</th>
<th>*</th>
<th>*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[[SN V] SN]</td>
<td>**</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>T</td>
<td>[SN [V SN]]</td>
<td>**</td>
<td>*</td>
</tr>
<tr>
<td>L’s target</td>
<td>“average”</td>
<td>2</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

The improved learning algorithm performs significantly better: Biró. ‘Towards a Robuster Interpretive Parsing: Learning from overt forms in Optimality Theory’. Submitted to *Journal of Logic, Language and Information*. 
Conclusions

The role of \textit{errors} = the results of imperfect mental computation.

- OT as a model of competence (static knowledge), Simulated Annealing for OT as a model of the (eventually erroneous) computation in the brain (performance).
- “Performance errors” as driving force behind language change.
- Language learning until convergence on performance patterns (measured using Jensen-Shannon Divergence).
- Different learning methods need different numbers of learning step until convergence.
- Learning despite hidden (covert) information.
Conclusions

The role of errors = the results of imperfect mental computation.

- OT as a model of competence (static knowledge), Simulated Annealing for OT as a model of the (eventually erroneous) computation in the brain (performance).
- “Performance errors” as driving force behind language change.
- Language learning until convergence on performance patterns (measured using Jensen-Shannon Divergence).
- Different learning methods need different numbers of learning step until convergence.
- Learning despite hidden (covert) information.
Conclusions

The role of *errors* = the results of imperfect mental computation.

- OT as a model of competence (static knowledge), Simulated Annealing for OT as a model of the (eventually erroneous) computation in the brain (performance).
- “Performance errors” as driving force behind language change.
  - Language learning until convergence on performance patterns (measured using Jensen-Shannon Divergence).
  - Different learning methods need different numbers of learning step until convergence.
  - Learning despite hidden (covert) information.
Conclusions

The role of *errors* = the results of imperfect mental computation.

- OT as a model of competence (static knowledge),
  Simulated Annealing for OT as a model of the (eventually erroneous) computation in the brain (performance).
- “Performance errors” as driving force behind language change.
- Language learning until convergence on performance patterns (measured using Jensen-Shannon Divergence).
  - Different learning methods need different numbers of learning step until convergence.
  - Learning despite hidden (covert) information.
Conclusions

The role of errors = the results of imperfect mental computation.

- OT as a model of competence (static knowledge), Simulated Annealing for OT as a model of the (eventually erroneous) computation in the brain (performance).
- “Performance errors” as driving force behind language change.
- Language learning until convergence on performance patterns (measured using Jensen-Shannon Divergence).
- Different learning methods need different numbers of learning step until convergence.
- Learning despite hidden (covert) information.
Conclusions

The role of *errors* = the results of imperfect mental computation.

- OT as a model of competence (static knowledge), Simulated Annealing for OT as a model of the (eventually erroneous) computation in the brain (performance).
- “Performance errors” as driving force behind language change.
- Language learning until convergence on performance patterns (measured using Jensen-Shannon Divergence).
- Different learning methods need different numbers of learning step until convergence.
- Learning despite hidden (covert) information.
Thank you for your attention!

Tamás Biró:
t.s.biro@uva.nl

Tools for Optimality Theory
http://www.birot.hu/OTKit/

Work supported by: