Agent-based modeling:
Agents with a complex cognitive architecture
(A tutorial)

Tamás Biró

Eötvös Loránd University (ELTE)

Network Theory and Computer Modeling in the Study of Religion
September 1, 2016
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></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).
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).
**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 sege</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).**
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 seqe</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).
Sentential negation (Jespersen’s cycle)

Research questions:
- Why does this language change happen?
- What drives change?
  “Performance errors” as a driving force behind language change?

Methodology:
- Multi-agent simulations

Reference
Learning (what CSR misses from CogSci)

● **Learning**: the algorithm behind / modeling acquisition. A central topic in linguistics and cognitive science, but missing in CSR. See also *machine learning.*
Errors of the mental computation

COMPETENCE

PERFORMANCE

static knowledge processes in the brain
Errors of the mental computation

COMPETENCE

PERFORMANCE

static knowledge

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

COMPETENCE → PERFORMANCE

LEARNING

COMPETENCE → PERFORMANCE
The *Iterative Learning Model*

- **Learning**: the algorithm behind / modeling acquisition. A central topic in linguistics and cognitive science, but missing in CSR. See also *machine learning*.

- **Iterative learning**:

  \[
  \text{Gen-0} \rightarrow \text{Gen-1} \rightarrow \text{Gen-2} \rightarrow \text{Gen-3}
  \]

- Simon Kirby at al.: language evolution (in biological evolution’s timescale). “Learning bottleneck” creates linguistic structure.

- **Others**: language change (in historical timescale). Assumption: language change takes place from generation to generation, due to imperfect acquisition. (Only partly true.)
(Possible) components of the model

Who learns from whom?

- $N$ agents in **one generation**.

- **Series of generations**: language produced by agents in Generation $k$ used as learning data by agents in Gen. $k + 1$.

\[
\begin{align*}
\text{Generation } k: & & a_1 & a_2 & a_3 & a_4 \\
\downarrow & \times & \times & \downarrow & \times & \downarrow \\
\text{Generation } k + 1: & & b_1 & b_2 & b_3 & b_4
\end{align*}
\]

- Note the strict intergenerational structure: no learning from grandparents, elder siblings or peers.

- Social structure? More learning data from parents? Learning data with more weight from people with prestige?
(Possible) components of the model

*Who learns from whom?*

- **$N$ agents in one generation.**
- **Series of generations:** language produced by agents in Generation $k$ used as learning data by agents in Gen. $k + 1$.

  Generation $k$: $a1 \ a2 \ a3 \ a4$

  Generation $k + 1$: $b1 \ b2 \ b3 \ b4$

- Note the strict intergenerational structure: no learning from grandparents, elder siblings or peers.
- Social structure? More learning data from parents? Learning data with more weight from people with prestige?
(Possible) components of the model

Who learns at all?

An agent composed of:

- **Knowledge**: a.k.a. competence, grammar, etc.
  Here: Optimality Theory (Prince and Smolensky 1993/2006)

- **Production**: a.k.a. performance, etc.
  Here: Simulated Annealing for Optimality Theory (Biró 2006)

- **Learning**: a.k.a. acquisition, etc.
  Here: online learning algorithms for Optimality Theory (Boersma and Hayes 2001; Magri 2012)
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 = $\mathcal{E}$ (globally) optimal candidate

SA-OT implementation performance model
produced forms = globally or locally optimal candidates
Sentential negation: Jespersen’s cycle

**Generation 1:**
- Competence: grammatical form is [SN V].
  Grammar: *NEGATION ≫ NEGATIONFIRST ≫ NEGATIONLAST
- Performance: 100% [SN V].

**Generation 2** learning from performance pattern of Generation 1:
- Competence: grammatical form is [SN V].
  Grammar: NEGATIONFIRST ≫ *NEGATION ≫ NEGATIONLAST
- Performance: 90% [SN V], and 10% [SN [V SN]].

**Generation 3** learning from performance pattern of Generation 2. Etc.
Sentential negation: Jespersen’s cycle

Generation 1:

- Competence: grammatical form is [SN V].
  Grammar: \( \neg \text{NEGATION} \Rightarrow \neg \text{NEGATION-FIRST} \Rightarrow \neg \text{NEGATION-LAST} \)
- Performance: 100% [SN V].

Generation 2 learning from performance pattern of Generation 1:

- Competence: grammatical form is [SN V].
  Grammar: \( \neg \text{NEGATION-FIRST} \Rightarrow \neg \text{NEGATION} \Rightarrow \neg \text{NEGATION-LAST} \)
- Performance: 90% [SN V], and 10% [SN [V SN]].

Generation 3 learning from performance pattern of Generation 2. Etc.
Sentential negation: Jespersen’s cycle

**Generation 1:**
- Competence: grammatical form is [SN V].
  - Grammar: \( *\text{NEGATION} \gg \text{NEGATIONFIRST} \gg \text{NEGATIONLAST} \)
- Performance: 100% [SN V].

**Generation 2** learning from performance pattern of Generation 1:
- Competence: grammatical form is [SN V].
  - Grammar: \( \text{NEGATIONFIRST} \gg *\text{NEGATION} \gg \text{NEGATIONLAST} \)
- Performance: 90% [SN V], and 10% [SN [V SN]].

**Generation 3** learning from performance pattern of Generation 2. Etc.
Sentential negation: Jespersen’s cycle

**Generation 1:**
- Competence: grammatical form is [SN V].
  Grammar: *NEGATION ≫ NEGATIONFIRST ≫ NEGATIONLAST
- Performance: 100% [SN V].

**Generation 2** learning from performance pattern of Generation 1:
- Competence: grammatical form is [SN V].
  Grammar: NEGATIONFIRST ≫ *NEGATION ≫ NEGATIONLAST
- Performance: 90% [SN V], and 10% [SN [V SN]].

**Questions:**
1. Has the learning been successful?  
   1. grammaticality judgement; 2. grammar; 3. performance pattern.
2. A way to model diachronic change?
3. Learner hears “SN V SN”: is it [[SN V] SN] or [SN [V SN]]?
Sentential negation: Jespersen’s cycle

**Generation 1:**
- Competence: grammatical form is [SN V].
  - Grammar: \( \text{NEGATION} \gg \text{NEGATIONFIRST} \gg \text{NEGATIONLAST} \)
- Performance: 100% [SN V].

**Generation 2** learning from performance pattern of Generation 1:
- Competence: grammatical form is [SN V].
  - Grammar: \( \text{NEGATIONFIRST} \gg \text{*NEGATION} \gg \text{NEGATIONLAST} \)
- Performance: 90% [SN V], and 10% [SN [V SN]].

**Questions:**

1. Has the learning been successful?
   1. grammaticality judgement; 2. grammar; 3. performance pattern.

2. A way to model diachronic change?

3. Learner hears “SN V SN”: is it [[SN V] SN] or [SN [V SN]]?
Sentential negation: Jespersen’s cycle

Generation 1:
- Competence: grammatical form is \([\text{SN V}]\).
  Grammar: \(*\text{NEGATION} \gg \text{NEGATIONFIRST} \gg \text{NEGATIONLAST}\)
- Performance: 100\% \([\text{SN V}]\).

Generation 2 learning from performance pattern of Generation 1:
- Competence: grammatical form is \([\text{SN V}]\).
  Grammar: \(\text{NEGATIONFIRST} \gg *\text{NEGATION} \gg \text{NEGATIONLAST}\)
- Performance: 90\% \([\text{SN V}]\), and 10\% \([\text{SN [V SN]}]\).

Questions:
1. Has the learning been successful?
   1. grammaticality judgement; 2. grammar; 3. performance pattern.
2. A way to model diachronic change?
3. Learner hears “\text{SN V SN}”: is it \([\text{[SN V] SN]}\) or \([\text{SN [V SN]}]\)?
## Modelling linguistic competence

**Faith[Neg] >> *Negation >> NegationFirst >> 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>
</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>
<tr>
<td>...</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
grammatical form = $\mathfrak{E}$ (globally) optimal candidate

SA-OT
implementation performance model
produced forms = globally or locally optimal candidates
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

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

Locally optimal forms: \(\blackdiamondsuit [\text{SN V}] \) 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.
<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.
<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.
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”:


Conclusions

What is the question / interest:

- **Proceed from the phenomenon:** explaining Jespersen’s cycle.

- **Proceed from theory:** role of \( \text{errors} = \text{results of imperfect mental computation in language change.} \)

- **Proceed from framework:** the behavior of a certain theoretical, computational, mathematical framework.
Conclusions

Model:

- Agents $\rightarrow$ (un)structured population $\rightarrow$ generation.

- Agents $\rightarrow$ knowledge (competence), production (performance) and learning (acquisition).

- Iterative learning model
Practicalities:

- Developed in own software *OTKit* ([http://www.birot.hu/OTKit/](http://www.birot.hu/OTKit/)).

- The more complex a model: the more parameters.

- The convincing force of a complex, still abstract and oversimplified computational model?
Thank you for your attention!

Tamás Biró:
tamas.biro@btk.elte.hu

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

Work supported by: