Language Acquisition
Fall 2010/Winter 2011

Morphology & Syntax

Afra Alishahi, Heiner Drenhaus
Computational Linguistics and Phonetics
Saarland University
Rules that Govern Form

• Moving from fixed forms (e.g. ‘apple’) to derivational forms

  play → plays, played, playing
  I, you, admire → “I admire you”

• Morphology and syntax
  • In all languages, the formation of words and sentences follows highly regular patterns
  • How are the regulations and exceptions represented?

• The study and analysis of language production in children reveals common and persistent patterns
U-shaped Learning Curves

• Observed U-shaped learning curves in children
  • **Imitation:** an early phase of conservative language use
  • **Generalization:** general regularities are applied to new forms
  • **Overgeneralization:** occasional misapplication of general patterns
  • **Recovery:** over time, overgeneralization errors cease to happen

• **Lack of Negative Evidence**
  • Children do not receive reliable corrective feedback from parents to help them overcome their mistakes (Marcus, 1993)
Case Study: Learning English Past Tense

• The problem of English past tense formation:
  • Regular formation: stem + ‘ed’
  • Irregulars do show some patterns
    • No-change: hit → hit
    • Vowel-change: ring → rang, sing → sang

• Over-regularizations are common: goed
  • These errors often occur after the child has already produced the correct irregular form: went

• What causes the U-shaped learning curve?
A Symbolic Account of English Past Tense

- Dual-Route Account: two qualitatively different mechanisms

- Prediction:
  - Errors result from transition from rote learning to rule-governed
  - Recovery occurs after sufficient exposure to irregulars
A Connectionist Account of Learning English Past Tense

• A connectionist model (Plunkett & Marchman, 1993)

Output units: phonological features of past tense

hidden units

Input units: phonological features of the stem

• Properties:

• Early in training, the model shows tendency to overgeneralize; by the end of training, it exhibits near perfect performance

• U-shaped performance is achieved using a single learning mechanism, but depends on sudden change in the training size
A Hybrid, Analogy-based Account

- Taatgen & Anderson (2002): an rational model of learning past tense based on the ACT-R architecture
  - Declarative memory chunks represent past tenses, both as a goal and as examples

```
PAST-TENSE-GOAL23
ISA PAST
OF WALK
STEM NIL
SUFFIX NIL
```

goal to determine past tense of walk

```
PAST-TENSE-GOAL23
ISA PAST
OF WALK
STEM WALK
SUFFIX ED
```

accomplished goal, stored in the memory
A Hybrid, Analogy-based Account

- The analogy strategy is implemented by two production rules, based on simple pattern matching:

**RULE ANALOGY-FILL-SLOT**

**IF** the goal has an empty *suffix* slot

**AND** there is an example in which *suffix* has a value

**THEN** set the *suffix* of the goal to the *suffix* value of the example

**RULE ANALOGY-COPY-A-SLOT**

**IF** the goal has an empty *stem* slot and the *of* slot has a certain value

**AND** in the example the values of the *of* and *stem* slots are equal

**THEN** set the *stem* to the value of the *of* slot
Hahn and Nakisa (2000) model, the network is trained on a partial vocabulary and is tested on the rest of it. The model did very well on unknown words, being correct 81% of the time. This indicates that information is contained within the phonological structure of the word, enabling the model to often guess the right inflection correctly. Nevertheless the model does not learn to apply the -s default rule. Instead, Hahn and Nakisa challenge the Marcus et al. (1995) claim that German has a default rule at all. The fact remains though, that German speakers use the -s suffix much more often than their model.

### ACT-R Architecture

The basic theoretical foundation of the ACT-R architecture is rational analysis (Anderson, 1990). According to rational analysis, each component of the cognitive system is optimized with respect to demands from the environment, given its computational limitations. The main components in ACT-R are a declarative (fact) memory and a production (rule) memory. To avoid confusion with grammatical rules, we will refer to rules in production memory with production rules. ACT-R is a so-called hybrid architecture, in the sense that it has both symbolic and sub-symbolic aspects. We will introduce these components informally. Table 1 provides a formal specification of some critical aspects of ACT-R equations.

<table>
<thead>
<tr>
<th>Equation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Activation</strong></td>
<td>The activation of a chunk has three parts: base-level activation, spreading activation from the current context and noise. Since spreading activation is a constant factor in the models discussed, we treat activation as if it were just base-level activation.</td>
</tr>
<tr>
<td>( A = B + \text{context} + \text{noise} )</td>
<td></td>
</tr>
<tr>
<td><strong>Base-level activation</strong></td>
<td>( B(t) = \log \sum_{j=1}^{n} (t - t_j)^{-d} )</td>
</tr>
<tr>
<td>( n ) is the number of times a chunk has been retrieved from memory, and ( t_j ) represents the time at which each of these retrievals took place. So, the longer ago a retrieval was, the less it contributes to the activation. ( d ) is a fixed ACT-R parameter that represents the decay of base-level activation in declarative memory.</td>
<td></td>
</tr>
<tr>
<td><strong>Retrieval time</strong></td>
<td>Activation determines the time required to retrieve a chunk. ( A ) is the activation of the chunk that has to be retrieved, and ( F ) and ( f ) are fixed ACT-R parameters. Retrieval will only succeed as long as the activation is larger than retrieval threshold ( \tau ), which is also a fixed parameter.</td>
</tr>
<tr>
<td>Time = ( Fe^{-fA} )</td>
<td></td>
</tr>
<tr>
<td><strong>Expected outcome</strong></td>
<td>Expected outcome is based on three quantities, the estimated probability of success of a production rule ( P ), the estimated cost of the production rule ( C ), and the value of the goal ( G ).</td>
</tr>
<tr>
<td>Expected outcome = ( PpG - C_p + \text{noise} )</td>
<td></td>
</tr>
</tbody>
</table>
A Hybrid, Analogy-based Account

- ACT-R’s production rule mechanism learns new rules by combining two rules that have fired consecutively into one:

**RULE LEARNED-REGULAR-RULE**

**IF** the goal is to find the past tense of a word and slots *stem* and *suffix* are empty

**THEN** set the *suffix* slot to ED and set the *stem* slot to the word of which you want the past tense
A Hybrid, Analogy-based Account

Fig. 3. Results of the model. (a) Proportions of responses by the model over time. Incorrect regulars are not indicated since these are all "Regular not inflected". (b) Overregularization of the model as it is usually plotted: overregularization is equal to (irregular correct)/(irregular correct + irregular regularized), and regular mark rate equals (regular correct)/(regular correct + regular incorrect).
Innateness of Language

• Central claim: humans have innate knowledge of language
  • Assumption: all languages have a common structural basis

• Argument from the Poverty of the Stimulus (Chomsky 1965)
  • Linguistic experience of children is not sufficiently rich for learning the grammar of the language, hence they must have some innate specification of grammar
  • Assumption: knowing a language involves knowing a grammar

• Universal Grammar (UG)
  • A set of rules which organize language in the human brain
Principles & Parameters

• A framework for representing UG
  • A finite set of fundamental principles that are common to all languages
    • E.g., “a sentence must have a subject”
  • A finite set of parameters that determine syntactic variability amongst languages
    • E.g., a binary parameter that determines whether the subject of a sentence must be overtly pronounced
  • Learning involves identifying the correct grammar
    • I.e., setting UG parameters to proper values for the current language
Computational Implementation of P&P

• Formal parameter setting models for a small set of grammars

• General approach:
  • Analyze current input string and set the parameters accordingly
  • Set a parameter when receiving evidence from an example which exhibits that parameter (trigger)

• Representative models:
  • Triggering Learning Algorithm or TLA [Gibson & Wexler, 1994]
  • Structural Triggers Learner or STL [Fodor, 1998]
  • Variational Learner or VL [Yang, 2002]
Computational Implementation of P&P

- TLA: randomly modifies a parameter value if it cannot parse the input
- STL: learns sub-trees (treelets) as parameter values
- VL: assigns a weight to each parameter, and rewards or penalizes these weights depending on parsing success
Computational Implementation of P&P

- TLA: chooses one of the possible interpretations of the ambiguous trigger
- STL: ignores ambiguous triggers and waits for unambiguous ones
- VL: each interpretation is parsed and the parameter weights are changed accordingly

What if it is ambiguous?
Computational Challenges of P&P

• Practical limitations:
  • Formalizing a UG that covers existing languages is a challenge
  • Learning relies on well-formed sentences as input
  • P&P framework predicts a huge space of possible grammars
    • 20 binary parameters lead to > 1 million grammars
  • Search spaces for a grammar contain local maxima
    • I.e. learner may converge to an incorrect grammar
  • Most of the P&P models are psychologically implausible
    • They predict that a child may repeatedly revisit the same hypothesis or jump randomly around the hypothesis space
Usage-based Accounts of Language Acquisition

• **Main claims:**
  
  • Children learn language regularities from input alone, without guidance from innate principles
  
  • Mechanisms of language learning are not domain-specific

• **Verb Island Hypothesis** (Tomasello, 1992)
  
  • Children build their linguistic knowledge around individual items rather than adjusting general grammar rules they already possess
  
  • Children use cognitive processes to gradually categorize the syntactic structure of their item-based constructions

  • General-purpose cognitive tools are used for this purpose: imitation, analogy, structure mapping
Distributional Representation as an Alternative to Grammar

- Knowing a language is **not** equated with knowing a grammar
  - Knowledge of language is developed to perform communicative tasks of comprehension and production

- Neural networks for language representation and acquisition
  - Different levels of linguistic representation are emergent structures that a network develops in the course of learning
Case Study: Elman (1990)

• A model of leaning lexical classes and word order

Input units: 2-3 word sentences

Hidden units

Context units: A copy of the hidden units is kept as context

Output units: Network is trained to predict the next word as output
### Word Categories

<table>
<thead>
<tr>
<th>Category</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOUN-HUM</td>
<td>man, woman</td>
</tr>
<tr>
<td>NOUN-ANIM</td>
<td>cat, mouse</td>
</tr>
<tr>
<td>NOUN-INANIM</td>
<td>book, rock</td>
</tr>
<tr>
<td>NOUN-AGRESS</td>
<td>dragon, monster</td>
</tr>
<tr>
<td>NOUN-FRAG</td>
<td>glass, plate</td>
</tr>
<tr>
<td>NOUN-FOOD</td>
<td>cookie, break</td>
</tr>
<tr>
<td>VERB-INTRAN</td>
<td>think, sleep</td>
</tr>
<tr>
<td>VERB-TRAN</td>
<td>see, chase</td>
</tr>
<tr>
<td>VERB-AGPAT</td>
<td>move, break</td>
</tr>
<tr>
<td>VERB-PERCEPT</td>
<td>smell, see</td>
</tr>
<tr>
<td>VERB-DESTROY</td>
<td>break, smash</td>
</tr>
<tr>
<td>VERB-EAT</td>
<td>eat</td>
</tr>
</tbody>
</table>
### Templates for Sentence Generation

<table>
<thead>
<tr>
<th>WORD 1</th>
<th>WORD 2</th>
<th>WORD 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOUN-HUM</td>
<td>VERB-EAT</td>
<td>NOUN-FOOD</td>
</tr>
<tr>
<td>NOUN-HUM</td>
<td>VERB-PERCEPT</td>
<td>NOUN-INANIM</td>
</tr>
<tr>
<td>NOUN-HUM</td>
<td>VERB-DESTROY</td>
<td>NOUN-FRAG</td>
</tr>
<tr>
<td>NOUN-HUM</td>
<td>VERB-INTRAN</td>
<td></td>
</tr>
<tr>
<td>NOUN-HUM</td>
<td>VERB-TRAN</td>
<td>NOUN-HUM</td>
</tr>
<tr>
<td>NOUN-HUM</td>
<td>VERB-AGPAT</td>
<td>NOUN-INANIM</td>
</tr>
<tr>
<td>NOUN-HUM</td>
<td>VERB-AGPAT</td>
<td></td>
</tr>
<tr>
<td>NOUN-ANIM</td>
<td>VERB-EAT</td>
<td>NOUN-FOOD</td>
</tr>
<tr>
<td>NOUN-ANIM</td>
<td>VERB-TRAN</td>
<td>NOUN-ANIM</td>
</tr>
<tr>
<td>NOUN-ANIM</td>
<td>VERB-AGPAT</td>
<td>NOUN-INANIM</td>
</tr>
<tr>
<td>NOUN-ANIM</td>
<td>VERB-AGPAT</td>
<td></td>
</tr>
<tr>
<td>NOUN-INANIM</td>
<td>VERB-AGPAT</td>
<td></td>
</tr>
<tr>
<td>NOUN-AGRESS</td>
<td>VERB-DESTROY</td>
<td>NOUN-FRAG</td>
</tr>
<tr>
<td>NOUN-AGRESS</td>
<td>VERB-EAT</td>
<td>NOUN-HUM</td>
</tr>
<tr>
<td>NOUN-AGRESS</td>
<td>VERB-EAT</td>
<td>NOUN-ANIM</td>
</tr>
<tr>
<td>NOUN-AGRESS</td>
<td>VERB-EAT</td>
<td>NOUN-FOOD</td>
</tr>
</tbody>
</table>
### Sample Training Sequence

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000000000000000000000000000000010 (woman)</td>
<td>000000000000000000000000000000010000 (smash)</td>
</tr>
<tr>
<td>000000000000000000000000000000010000 (smash)</td>
<td>000000000000000000000000000000010000 (plate)</td>
</tr>
<tr>
<td>000000000000000000000000000000010000 (plate)</td>
<td>000000000000000000000000000000010000 (cat)</td>
</tr>
<tr>
<td>000000000000000000000000000000010000 (cat)</td>
<td>000000000000000000000000000000010000 (move)</td>
</tr>
<tr>
<td>000000000000000000000000000000010000 (move)</td>
<td>000000000000000000000000000000010000 (man)</td>
</tr>
<tr>
<td>000000000000000000000000000000010000 (man)</td>
<td>000100000000000000000000000000010000 (break)</td>
</tr>
<tr>
<td>000100000000000000000000000000010000 (break)</td>
<td>000010000000000000000000000000010000 (car)</td>
</tr>
<tr>
<td>000010000000000000000000000000010000 (car)</td>
<td>010000000000000000000000000000010000 (boy)</td>
</tr>
<tr>
<td>010000000000000000000000000000010000 (boy)</td>
<td>000000000000000000000000000000010000 (move)</td>
</tr>
<tr>
<td>000000000000000000000000000000010000 (move)</td>
<td>000000000000000000000000000000010000 (girl)</td>
</tr>
<tr>
<td>000000000000000000000000000000010000 (girl)</td>
<td>000000000000000000000000000000010000 (eat)</td>
</tr>
<tr>
<td>000000000000000000000000000000010000 (eat)</td>
<td>010000000000000000000000000000010000 (bread)</td>
</tr>
<tr>
<td>010000000000000000000000000000010000 (bread)</td>
<td>000000000010000000000000000000010000 (dog)</td>
</tr>
<tr>
<td>000000000010000000000000000000010000 (dog)</td>
<td>000000000000000000000000000000010000 (move)</td>
</tr>
<tr>
<td>000000000000000000000000000000010000 (move)</td>
<td>000000000000000000000000000000010000 (mouse)</td>
</tr>
<tr>
<td>000000000000000000000000000000010000 (mouse)</td>
<td>000000000000000000000000000000010000 (mouse)</td>
</tr>
<tr>
<td>000000000000000000000000000000010000 (mouse)</td>
<td>000000000000000000000000000000010000 (move)</td>
</tr>
<tr>
<td>000000000000000000000000000000010000 (move)</td>
<td>100000000000000000000000000000010000 (book)</td>
</tr>
<tr>
<td>100000000000000000000000000000010000 (book)</td>
<td>000000000000000000000000000000010000 (lion)</td>
</tr>
</tbody>
</table>
Analysis of Hidden Unit Activation Patterns

Figure 7. Hierarchical cluster diagram of hidden unit activation vectors in simple sentence prediction task. Labels indicate the inputs which produced the hidden unit vectors: inputs were presented in context, and the hidden unit vectors averaged across multiple contexts.

Several points should be emphasized. First, the category structure appears to be hierarchical. Thus, "dragons" are large animals, but also members of the class [\[...\]+animate] nouns. The hierarchical interpretation is achieved through the way in which the spatial relations (of the representations) are organized. Representations that are near one another in the representational space form classes, while higher level categories correspond to larger and more general regions of this space. Second, it is also true that the hierarchy is "soft" and implicit. While some categories may be qualitatively distinct (i.e., very far from each other...
Learning Grammar from Corpora

• Many computational models show the possibility of learning a grammar from corpus data
  • Machine learning techniques induce a grammar that fits data

• Common properties:
  • Most of these models are not incremental
  • They mostly focus on the acquisition of syntax (usually a CFG), but not semantics
Case Study: MOSAIC (Jones et al., 2000)

- MOSAIC (Model Of Syntax Acquisition In Children; Jones et al 2000)
  - Learns from raw text, and produces utterances similar to what children produce using a discrimination network
Case Study: MOSAIC (Jones et al., 2000)

- **Underlying mechanisms**
  - Learning: expand the network based on input data
  - production: traverse the network and output contents of the nodes

- **Generalization**
  - Generative links allow limited generalization abilities
  - Lack of semantic knowledge prevents meaningful generalization
  - Generalized sentences are limited to high-frequency terms

- **Evaluation**
  - The model was trained on a subset of CHILDES
  - It was used to simulate verb island phenomenon, optional infinitive in English, subject omission, ...