Learning by Analogy Versus Learning by Instruction: Same Knowledge, Different Representations

Timothy J. Nokes

Stellan Ohlsson & Andrew Corrigan-Halpern

Department of Psychology

University of Illinois at Chicago

Preview

- Background, Questions, and Goals of Research
- Empirical Study on Learning
  - Task
  - Method
  - Results
  - Interpretation
- Modeling Endeavor
  - ACT-R 4.0 Simulation
  - Design and Results
- Extensions

Multiple Paths to Knowledge

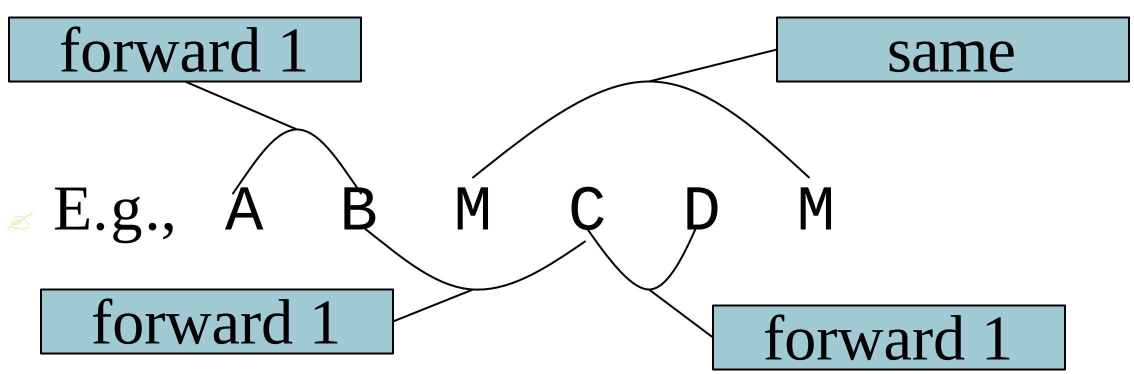
- Deep Knowledge
  - Abstract
  - Generative
- Theories of Knowledge Acquisition
  - Induction
  - Analogical Learning
  - Text Comprehension

General Questions

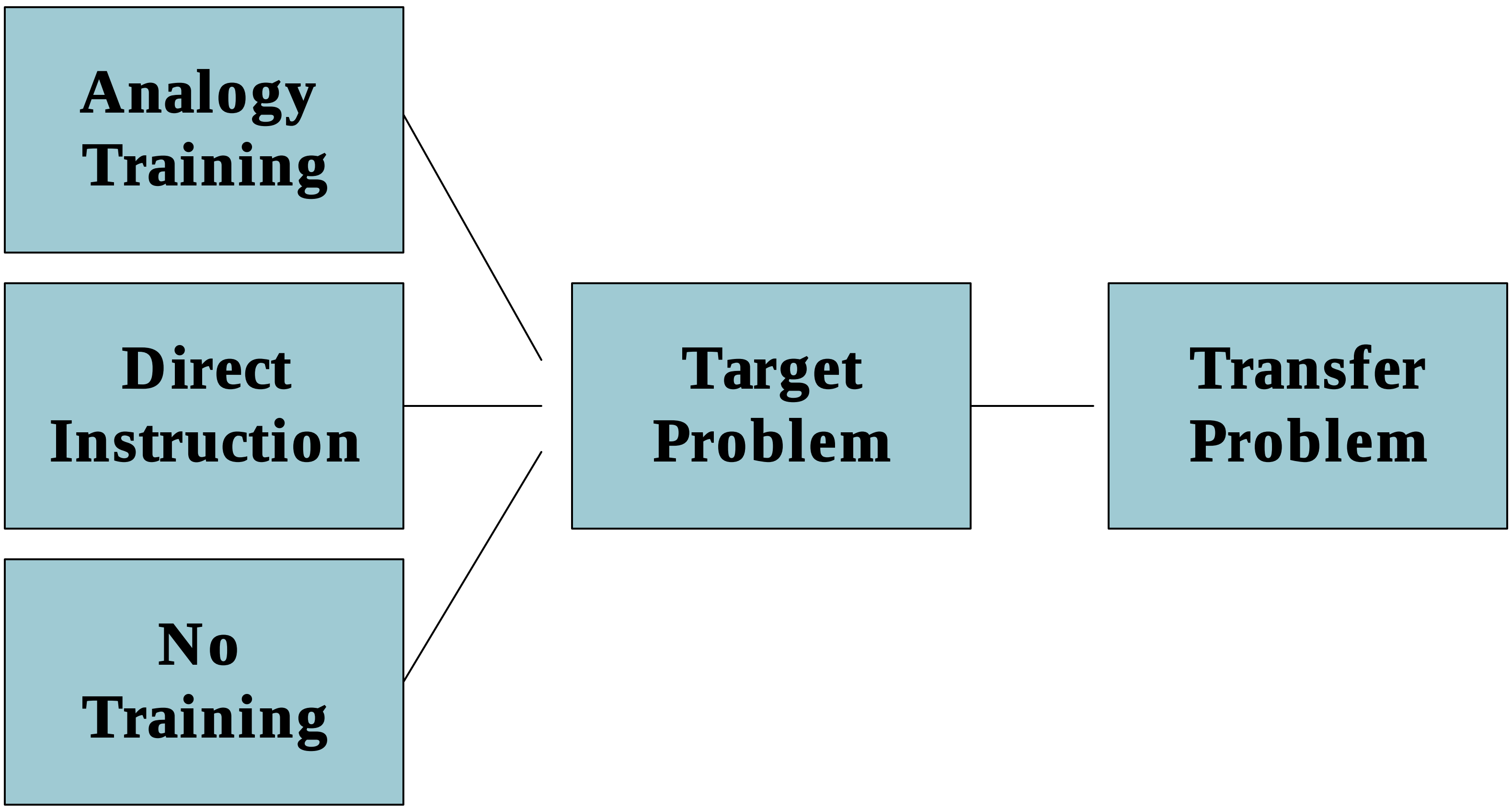
- What is learned from two particular learning situations: Analogy vs. Direct Instruction?
- Is the same knowledge acquired?
- Does performance differ on different measures or different tasks?

Empirical Study

- Target Task
  - Sequence Extrapolation Problem (Thurstone, 1949)
    - A B M C D M __ __ __ __
- Target Knowledge to be Learned
  - Specific pattern
    - Relations between letters

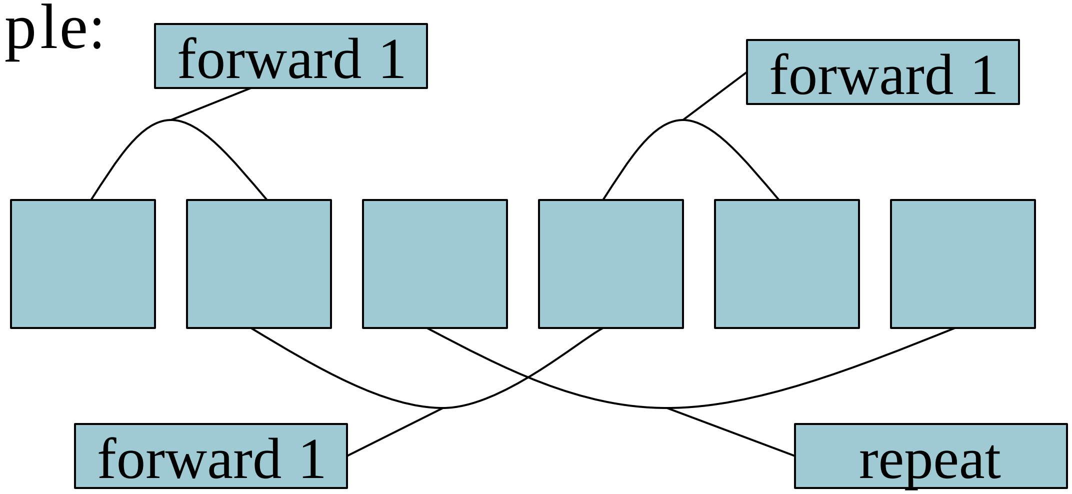


General Design

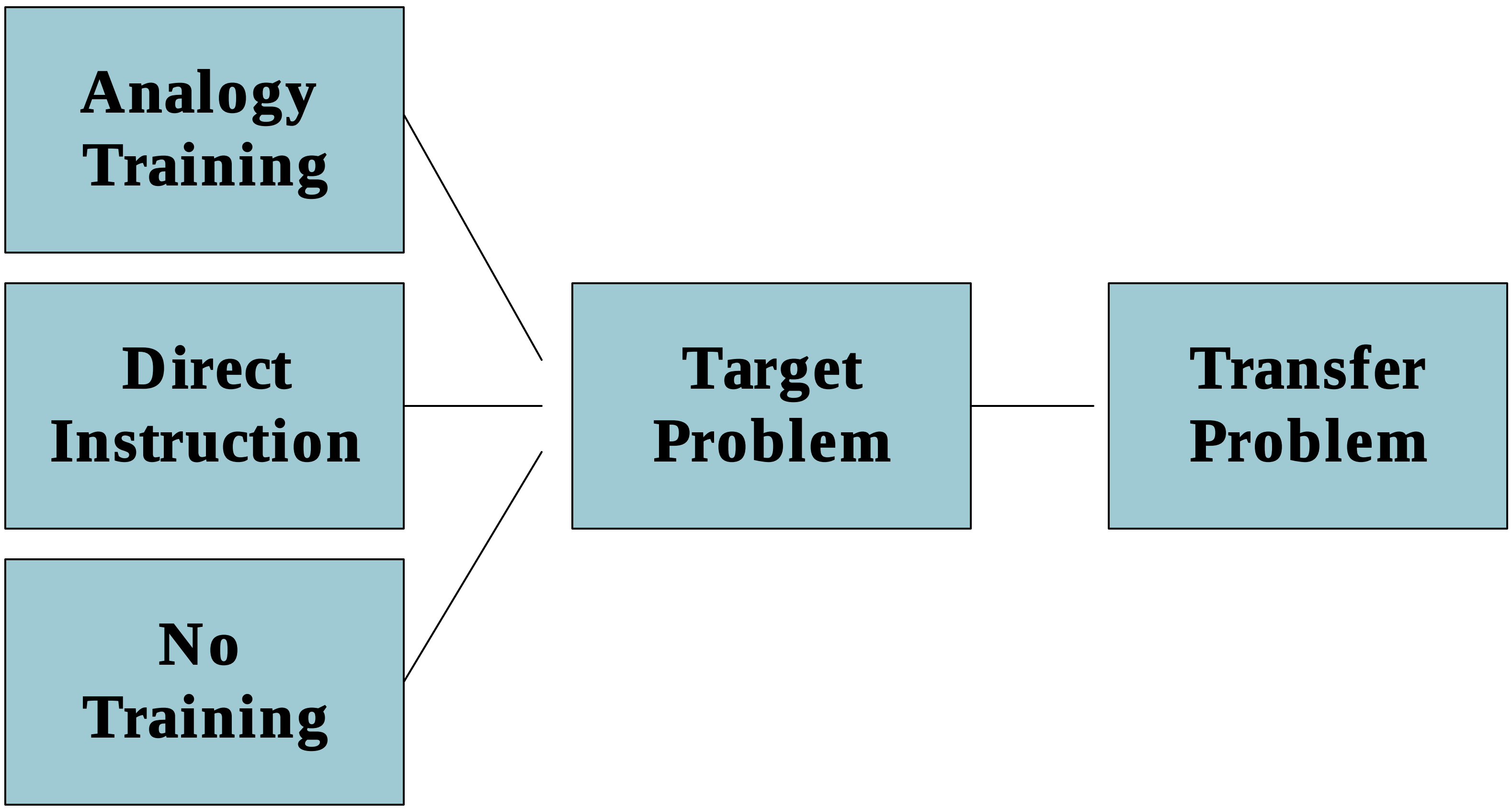


Method and Procedure

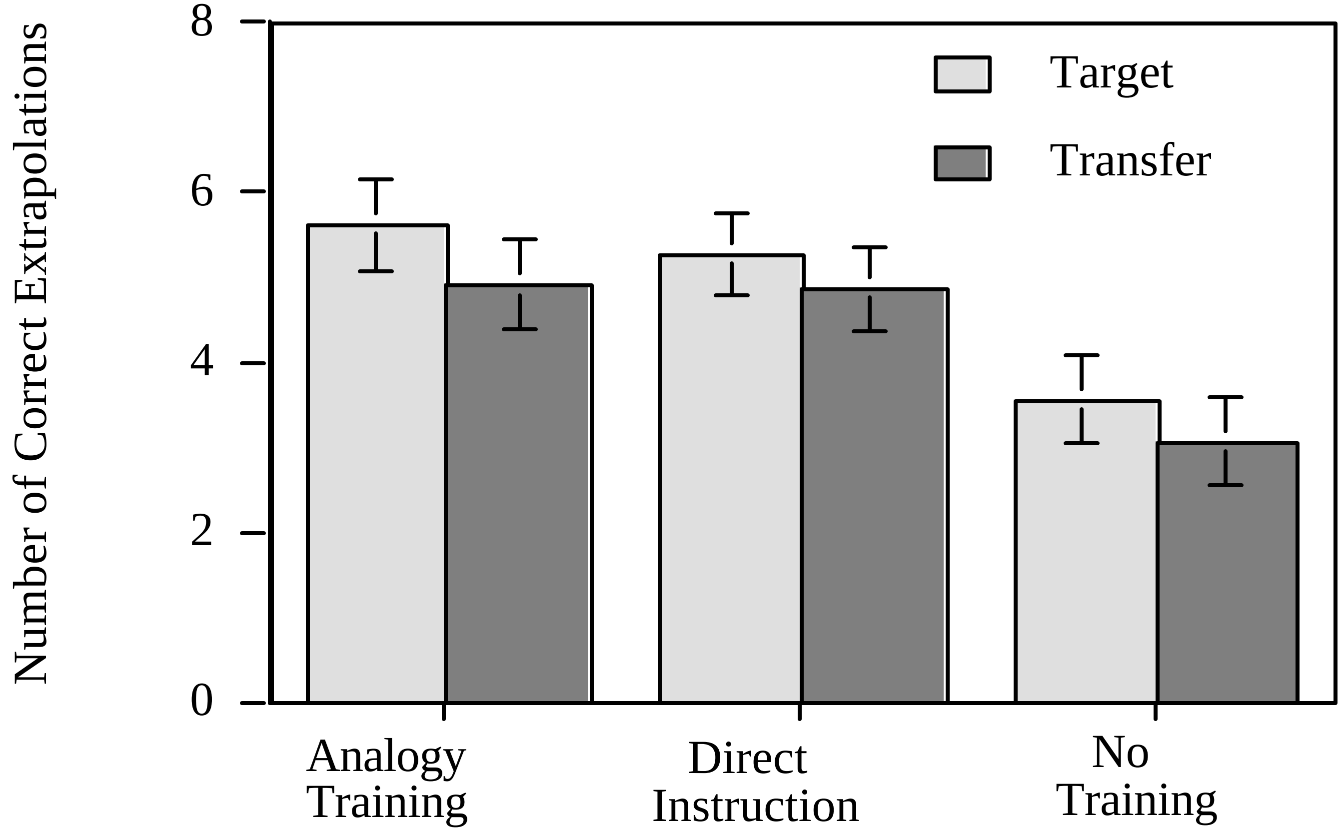
- Analogy groups solved similar problems
  - Same relations with different surface features
  - Example: `I J M K L M . . .`
- Direct Instruction
  - Read tutorials and memorized abstract patterns
  - Pattern Example:



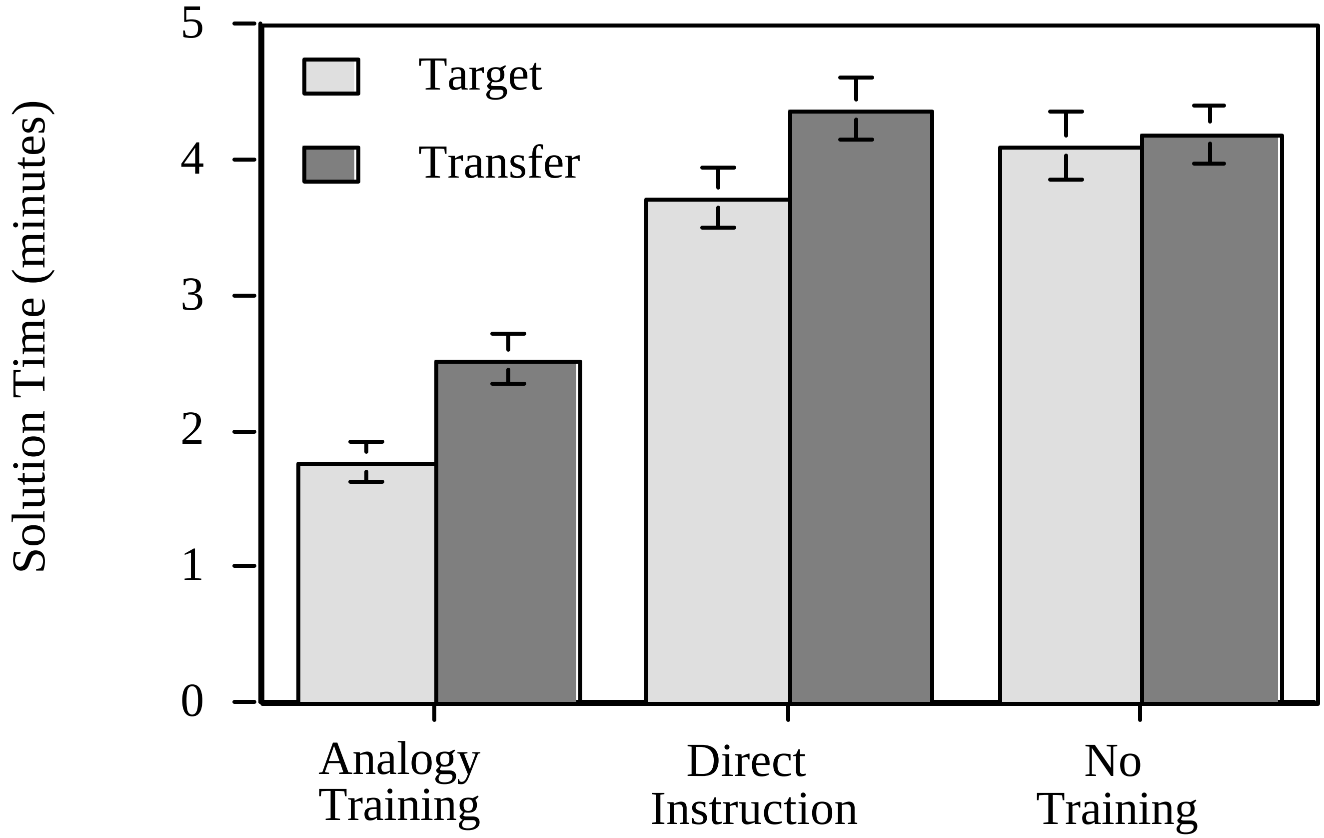
Prediction


Analogy Training
Direct Instruction
No Training
Target Problem
Transfer Problem

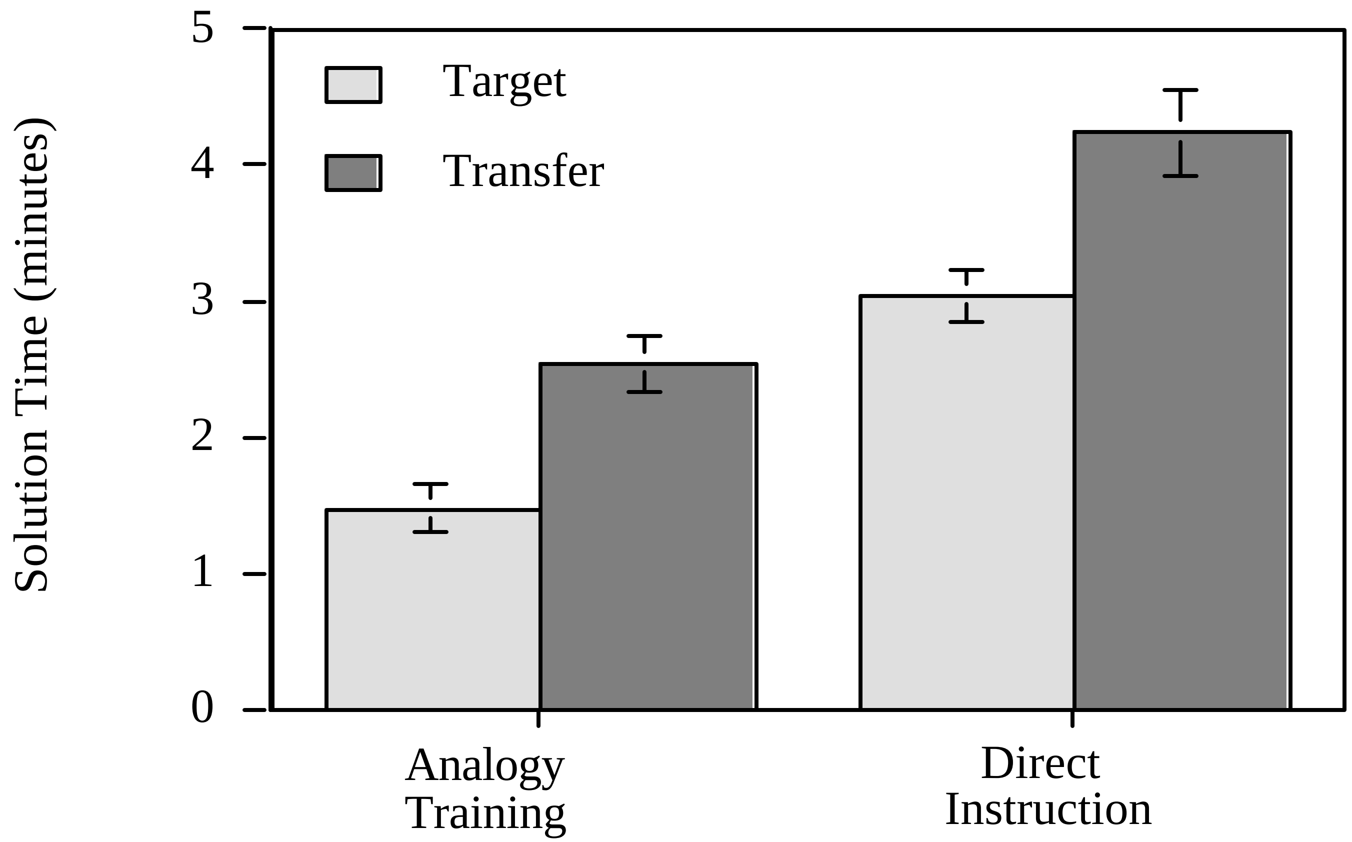
Empirical Results: Accuracy



Empirical Results: Solution Time



Solution Time for Perfect Performance



Summary of Empirical Results

- Same Accuracy Performance
  - Analogy = Direct Instruction
- Large Difference in Solution Times
  - Analogy < Direct Instruction
- This is true even when accuracy performance is near ceiling
  - Analogy < Direct Instruction

Interpretation

- What accounts for solution time differences?
- Hypothesis -
  - Different knowledge representations:
    - Analogy group acquired procedural knowledge of the target pattern via practice
    - Instruction group acquired declarative knowledge of the target pattern + general procedures for extrapolation

Simulation Models

- Terminal Models (e.g., Salvucci & Anderson, 1998)
  - Models of *what is learned* from each training scenario not models that learn
  - Analogy model and Instruction model
- Model Design
  - Knowledge common to both:
    - Alphabet and letter after and before relations
    - First two iterations of the to be solved problem
    - Position knowledge (‘position before and after’)
    - To be extrapolated positions (empty boxes)

Instruction Model

- Target pattern represented as declarative chunks:
  - Knowledge of the position relations
    - position 4 is one forward from position 2
  - Knowledge of analogous relations of the to be extrapolated positions
    - position 7 is like position 4
- Model had a general set of extrapolation interpretation production rules:
  - Identify relation ✍ Infer ‘relation’✍

    Identify empty position ✍ Fill in the Letter

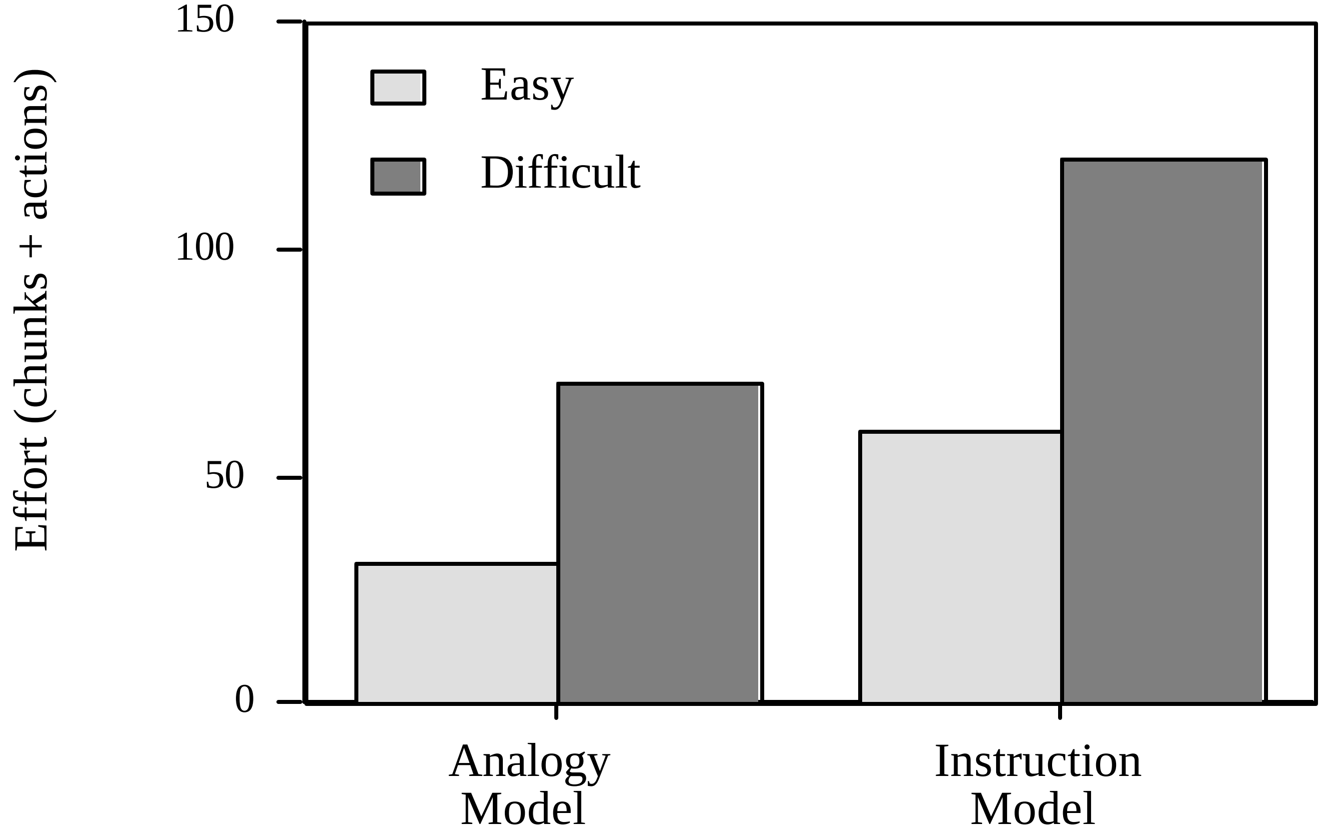
Analogy Model

- Target pattern encoded as a set of production rules.
  - Set of production rules tailored to each relation of the pattern.
  - Rules were of moderate abstraction in that they were not tied to particular letters, that is rules would fire for any given instance of the pattern regardless where it occurred in the alphabet.

Model Summary

- Instruction model has general extrapolation rules for interpreting specific declarative knowledge of the target pattern.
- Analogy model has specific extrapolation rules, the rules will only work for the target pattern and has limited declarative knowledge (just knowledge of the period).

Simulation Result



Results Summary

- Empirical
  - Accuracy performance the same between analogy and direct instruction.
  - However, Analogy group faster to solve the problems.
- Model
  - Analogy model takes less chunks and actions than the instruction model.
  - Predicts the qualitative difference between analogy training and direct instruction solution times.

Conclusion

- Both groups learned the target knowledge
- They represented that knowledge in different ways
- Analogy: procedural pattern specific knowledge via production compilation
- Instruction: declarative knowledge of the pattern
- Consequences for executing that knowledge

Extensions?

- Currently transforming the models into ACT-R 5.0
- Extend to Models of including the PM side of ACT-R
- Extend models to include pattern finding and learning
- Simulate fine details of performance including errors and RT on problems (trace data)
- Performance on transfer problems
- Soar vs. ACT-R??