Transfer of Knowledge in Cognitive Systems

Pat Langley
Computational Learning Laboratory
Center for the Study of Language and Information
Stanford University, Stanford, California USA
http://cll.stanford.edu/

Thanks to D. Aha, D. Choi, K. Forbus, J. Laird, S. Rogers, and T. Senator for useful discussions. This talk reports research funded by a grant from DARPA and AFRL, which are not responsible for its contents.
Humans exhibit general intelligence by their ability to learn in many domains.
Humans exhibit general intelligence by their ability to learn in many domains.

Humans are also able to utilize knowledge learned in one domain in other domains.
What is Transfer?

A learner exhibits transfer of learning from task/domain A to task/domain B when, after it has trained on A, it shows improved behavior on B.

- **learning curve for task A**
- **better intercept on task B**
- **faster learning rate on task B**
- **better asymptote on task B**
What is Transfer?

- Transfer is a *sequential* phenomenon that occurs in settings which involve on-line learning.
  - Thus, multi-task learning does not involve transfer.
- Transfer involves the reuse of knowledge *structures*.
  - Thus, it requires more than purely statistical learning.
- Transfer can lead to improved behavior (*positive* transfer).
  - But it can also produce worse behavior (*negative* transfer).
- Transfer *influences* learning but is not a *form* of learning.
  - Thus, “transfer learning” is an oxymoron, much like the phrase “learning performance”.
Roots of Transfer in Psychology

The notion of transfer comes from psychology, where it has been studied for over a hundred years:

- benefits of Latin (Thorndike & Woodworth, 1901)
- puzzle solving (Luchins & Luchins, 1970)
- operating devices (Kieras & Bovair, 1986)
- using text editors (Singley & Anderson, 1988)
- analogical reasoning (Gick & Holyoak, 1983)

Some recent studies have included computational models that predict the transfer observed under different conditions.
Inference tasks that require multi-step reasoning to obtain an answer, such as solving physics word problems and aptitude/achievement tests.

Classification tasks that involve assigning items to categories, such as recognizing types of vehicles or detecting spam. These are not very interesting.

Procedural tasks that involve execution of routinized skills, both cognitive (e.g., multi-column arithmetic) and sensori-motor (e.g., flying an aircraft).

Problem-solving tasks that benefit from strategic choices and heuristic search, such as complex strategy games.
The degree of transfer depends on the structure shared with the training tasks.

Transfer requires that knowledge be represented in a modular fashion.

Transfer requires the ability to *compose* these knowledge elements dynamically.

The degree of transfer depends on the *structure shared* with the training tasks.

Transfer across domains requires abstract relations among representations.
Dimensions of Knowledge Transfer

Knowledge transfer complexity is determined primarily by differences in the knowledge content and representation between the source and target problems.

- **We have not solved this before, but we know other pertinent information about this domain that uses the same representation.**

- **We have not solved similar problems, and are not familiar with this domain and problem representation.**

- **We know the solution to a similar problem with a different representation, possibly from another domain.**

- **We have already solved these problems.**

- **Similar Representations (e.g., within-domain transfer)**
  - **Knowledge Reuse**
  - **First-Principles Reasoning**
  - **Isomorphism**

- **Different Representations (e.g., most cross-domain transfer)**

**Problem Solver**
Memorization

Improvement in which the transfer tasks are the same as those encountered during training.

E.g., solving the same geometry problems on a homework assignment as were presented in class. This is not very interesting.
Within-Domain Lateral Transfer

Improvement on related tasks of similar difficulty within the same domain that share goals, initial state, or other structure.

E.g., solving new physics problems that involve some of the same principles but that also introduce new ones.
Within-Domain Vertical Transfer

Improvement on related tasks of greater difficulty within the same domain that build on results from training items.

E.g., solving new physics problems that involve the same principles but that also require more reasoning steps.
Cross-Domain Lateral Transfer

Improvement on related tasks of similar difficulty in a different domain that shares either higher-level or lower-level structures.

E.g., solving problems about electric circuits that involve some of the same principles as problems in fluid flow but that also introduce new ones.
Cross-Domain Vertical Transfer

Improvement on related tasks of greater difficulty in a different domain that share higher-level or lower-level structures.

E.g., solving physics problems that require mastery of geometry and algebra or applying abstract thermodynamic principles to a new domain.
Methods for cumulative learning of hierarchical skills and concepts define new cognitive structures in terms of structures learned on earlier tasks.

This approach is well suited to support vertical transfer to new tasks of ever increasing complexity.

Learning can operate on problem-solving traces, observations of another agent’s behavior, and even on direct instructions.
Methods for analogical reasoning store cognitive structures that encode relations in training problems.

Upon encountering a new problem, they retrieve stored experiences with similar relational structure.

Additional relations are then inferred based on elements in the retrieved problem.

Analogical reasoning can operate over any stored relational structure, but must map training elements to transfer elements, which can benefit from knowledge. This approach is well suited for lateral transfer to tasks of similar difficulty.
Approaches to Transfer: Mapping Representations

Source domain: Electricity

Knowledge: Ohm’s law
\[ I = \frac{V_1 - V_2}{R} \]

Mapping Process

Knowledge: Poiseuille’s law
\[ F = \frac{P_1 - P_2}{R} \]

Target domain: Fluid Flow

Transfer of learned knowledge across domains may require mapping between their representations of shared content.

Q: If \( P_1 = 3 \), \( P_2 = 2 \), and \( R = 2 \), then what force \( F \) is being applied, assuming we only know Ohm’s law for electric currents?
Experimental Studies of Transfer

Transfer condition
- Train on items from source domain
- Test and train on target domain items

Control condition
- Present no items from source domain
- Test and train on target domain items

Compare results from transfer and control conditions
Dependent Variables in Transfer Studies

Dependent variables for transfer experiments should include:

- Initial performance on the transfer tasks
- Asymptotic performance on the transfer tasks
- Rate of improvement on the transfer tasks

These require collecting learning curves over a series of tasks.

Such second-order variables build on basic metrics such as:

- Accuracy of response or solutions to tasks
- Speed or efficiency of solutions to tasks
- Quality or utility of solutions to tasks

Different basic measures are appropriate for different domains.
Transfer in Urban Combat

Urban Combat is a first-person shooter game developed at UT Arlington that we are using as a testbed to study approaches to the transfer of learned knowledge.
We are studying transfer in ICARUS, an architecture that incorporates some key assumptions from theories of human cognition:

1. Short-term memories are distinct from long-term stores
2. Memories contain modular elements cast as list structures
3. Long-term structures are accessed through pattern matching
4. Cognition occurs in retrieval/selection/action cycles
5. Performance and learning compose elements in memory

These claims give ICARUS much in common with other cognitive architectures like Soar and ACT-R.
ICARUS’ Functional Processes

- Long-Term Conceptual Memory
- Conceptual Inference
- Perceptual Buffer
- Short-Term Belief Memory
- Skill Retrieval and Selection
- Environment
- Skill Execution
- Long-Term Skill Memory
- Problem Solving/Skill Learning
- Motor Buffer
- Short-Term Goal Memory
Representing Long-Term Structures

ICARUS encodes two forms of general long-term knowledge:

- **Conceptual clauses**: A set of relational inference rules with perceived objects or defined concepts in their antecedents;

- **Skill clauses**: A set of executable skills that specify:
  - a head that indicates a goal the skill achieves;
  - a single (typically defined) precondition;
  - a set of ordered subgoals or actions for achieving the goal.

These define a specialized class of *hierarchical task networks* in a syntax very similar to Nau et al.’s SHOP2 formalism.

Beliefs, goals, and intentions are instances of these structures.
Primitive Concepts for Urban Combat

((stopped ?self)
 :tests (<= (+ (* ?xvel ?xvel) (* ?yvel ?yvel)) 1)))

((in-region ?self ?region)
 :percepts ((self ?self region ?region)))

((connected-region ?target ?gateway)
 :percepts ((gateway ?gateway region ?target)))

((blocked-gateway ?gateway)
 :percepts ((gateway ?gateway visible1 ?v1 visible2 ?v2))
 :tests ((and (equal ?v1 'B) (equal ?v2 'B))))

((first-side-blocked-gateway ?gateway)
 :percepts ((gateway ?gateway type ?type visible1 ?v1 visible2 ?v2))
 :tests ((equal ?type 'WALK) (equal ?v1 'B) (equal ?v2 'C)))

((flag-captured ?self flag1)
 :percepts ((self ?self holding ?flag1) (entity ?flag1))
 :tests ((not (equal ?flag NIL)))))
Nonprimitive Urban Combat Concepts

((not-stopped ?self)
 :percepts  ((self ?self))
 :relations  ((not (stopped ?self)))))

((clear-gateway ?gateway)
 :percepts  ((gateway ?gateway type ?type visible1 ?v1 visible2 ?v2))
 :relations  ((not-stopped ?self))
 :tests  ((equal ?type 'WALK) (equal ?v1 'C) (equal ?v2 'C)))))

((stopped-in-region ?self ?region)
 :percepts  ((self ?self))

((crossable-region ?target)
 :percepts  ((self ?self) (region ?target))
 :relations  ((connected-region ?target ?gateway) (clear-gateway ?gateway)))))

((in-region-able ?self ?current ?region)
Primitive Skills for Urban Combat

((in-region ?self ?region)
 :percepts (\(\text{self} ?\text{self}\)) (region ?current) (region ?region)
 (gateway ?gateway region ?region dist1 ?dist1 angle1 ?angle1
 dist2 ?dist2 angle2 ?angle2))
 :start (\(\text{in-region-able} ?\text{self} ?\text{current} ?\text{region}\))
 :actions (\(*\text{move-toward} (\text{max} ?\text{dist1} ?\text{dist2}) (\text{mid-direction} ?\text{angle1} ?\text{angle2})\))

((clear-gateway ?gateway)
 :percepts (\(\text{self} ?\text{self}\)) (gateway ?gateway type WALK))
 :start (\(\text{stopped} ?\text{self}\))
 :actions (\(*\text{move-toward} 50 0\))

((clear-gateway ?gateway)
 :percepts (gateway ?gateway region ?region dist1 ?dist1 angle1 ?angle1
 visible1 ?v1 dist2 ?dist2 angle2 ?angle2 visible2 ?v2))
 :start (\(\text{first-side-blocked-gateway} ?\text{gateway}\))
 :actions (\(*\text{move-toward} ?\text{dist2} ?\text{angle2}\))

((flag-captured ?self ?flag)
 :percepts (\(\text{self} ?\text{self}\)) (entity ?flag dist ?dist angle ?angle))
 :start (\(\text{in-region} ?\text{self region107}\))
 :actions (\(*\text{move-toward} ?\text{dist} ?\text{angle}\))
Nonprimitive Skills for Urban Combat

((crossable-region ?target)
 :percepts ((region ?target) (gateway ?gateway))
 :start ((connected-region ?target ?gateway))
 :subgoals ((clear-gateway ?gateway)))

((in-region-able ?self ?current ?region)

((stopped-in-region ?self ?region)
 :percepts ((self ?self) (region ?region))

((stopped-in-region ?self ?current)
 :percepts ((self ?self))
 :start ((in-region ?self ?current))
 :subgoals ((stopped ?self)))

((flag-captured ?self ?flag)
 :percepts ((self ?self))
Route Knowledge for Urban Combat

((in-region ?self region105)
 :percepts ((self ?self))

((in-region ?self region114)
 :percepts ((self ?self))
 :subgoals ((in-region-able ?self region105 region114) (in-region ?self region114)))

((in-region ?self region110)
 :percepts ((self ?self))

((in-region ?self region116)
 :percepts ((self ?self))

((in-region ?self region107)
 :percepts ((self ?self))
Hierarchical Structure of Long-Term Memory

ICARUS organizes both concepts and skills in a hierarchical manner.

Each concept is defined in terms of other concepts and/or percepts.

Each skill is defined in terms of other skills, concepts, and percepts.
ICARUS interleaves its long-term memories for concepts and skills.

For example, the skill highlighted here refers directly to the highlighted concepts.
Basic ICARUS Processes

ICARUS matches patterns to recognize concepts and select skills.

- Concepts are matched bottom up, starting from percepts.
- Skill paths are matched top down, starting from intentions.
Transfer in ICARUS

• What forms of knowledge does ICARUS transfer?
  • *Hierarchical/relational skill and concept clauses*

• Where does the transferred knowledge originate?
  • *It comes from experience on source problems and background knowledge*

• How does ICARUS know what to transfer?
  • *Skills are indexed by goals they achieve, with preference for more recently learned structures*
A Transfer Scenario from Urban Combat

Source Problem

Target Problem

Here the first part of the source route transfers to the target, but the second part must be learned to solve the new task.
Structures Transferred in Scenario

Source

Target

Shared structures
Other Transfer Results with ICARUS

We have also tested ICARUS in domains like FreeCell solitaire.

Experiments suggest that learned knowledge transfers well here.
Key Ideas about Transfer in ICARUS

• The most important transfer concerns goal-directed behavior that involves sequential actions aimed toward an objective.

• Transfer mainly involves the reuse of knowledge structures.

• Organizing structures in a hierarchy aids reuse and transfer.

• Indexing skills by goals they achieve determines relevance.

• One can learn hierarchical, relational, goal-directed skills by analyzing traces of expert behavior and problem solving.

• Skill learning can build upon structures acquired earlier.

• Successful transfer benefits from knowledge-based inference to recognize equivalent situations.
General Game Playing

GGP is a Web-based software environment developed at Stanford that supports:

• logical specification of many different games in terms of:
  • relational descriptions of states
  • legal moves and their effects
  • goal relations and their payoffs

• management of matches between automated players

• competitions that involve many players and games

The GGP framework (http://games.stanford.edu) encourages research on systems that exhibit *general* intelligence.

This summer, AAAI will host its second GGP competition.
Transfer in General Game Playing

GGP is especially appropriate for research on transfer because:

• one can specify game environments and rules precisely
• one can formalize relationships between different games
  • to support different types/levels of transfer
  • including ones that involve representational shifts
• the game manager eases running of experiments

We are using GGP to evaluate methods for transferring learned knowledge to new but related settings.
A GGP Scenario for Extrapolation Transfer

**Goal:** White king and rook in castle formation

**Rules:** Standard Chess

**Concepts:**
- castle-blocked(\(x, y\))
- castle-threatened(\(x, y\))
- king-in-check(\(player, x, y\))

---

**Source**

- White king and rook
- Black pawns and knight

**Target**

- White king and rook
- Black pawns and knight

Open Research Problems

There remain many research issues that we must still address:

• Goal transfer - across tasks with distinct but related objectives
• Negative transfer - minimizing use of inappropriate knowledge
• Context handling - avoiding catastrophic interference
• Representation mapping
  • Lateral - Deep analogy that involves partial isomorphisms
  • Vertical - Bootstrapped learning that builds on lower levels

These challenges should keep our field occupied for some time.
Closing Remarks

Transfer of learned knowledge is an important capability that:

• involves the sequential reuse of knowledge structures
• takes many forms depending on source/target relationships
• has been repeatedly examined within psychology/education
• has received little attention in AI and machine learning
• requires a fairly sophisticated experimental method

Transfer originated in psychology, and it is best studied in the context of cognitive architectures, which have similar roots.
End of Presentation