Bootstrapped Learning

Creating the Electronic Student that learns from Natural Instruction

Dan Oblinger
May 22, 2006
July 5, 2006

( Approved for Public Release, Distribution Unlimited )
A new focus for ML

**MACHINE LEARNING** is primarily a *modeling* tool. Used to build models when we know something, *but not everything relevant*, about some target problem.

**HUMAN MENTORED LEARNING** is primarily a *communication* tool. Used to communicate capabilities from an instructor, *who generally is assumed to have all relevant capabilities*, to a student that does not.

**BOOTSTRAP LEARNING** is a program to build an *electronic student*. Like its human counterpart, and unlike most ML of today, the electronic student assumes all relevant capabilities are possessed by the teacher, and the goal is to learn using the “same” instruction methods used between humans.

BL program develops new learning algorithms. Each algorithm is not specialized to a particular problem domain, instead they are specialized to a particular interaction with the teacher (to a particular “Natural Instruction” method).
Red team builds ladders – Blue Teams *cannot* construct custom solutions!
**BL Program Objective**

**Natural Instruction**: Methods humans use to instruct others

**Claim #1**: Bootstrap Learning (BL) will learn a wide range of performance tasks based on abstracted Natural Instruction (with zero reprogramming between performance tasks.)

**Claim #2**: Bootstrap Learning will compare with human learning given the “same” instructional materials, and the “same” background knowledge. (both senses of “same” are made precise by this program.)

**Field-Trainable-Systems are a key missing capability.**

Systems today cannot be specialized to particular battlespace because:

- Can’t account for every contingency in advance
- Rate of change (in mission, in enemy tactics, etc.) is too great to accommodate using traditional software update cycle.

We believe any military hardware with a CPU, sensors, and actuators, should be field trainable.

A *general-purpose* “electronic student” that bootstraps complex behaviors
## Ladder API – input for Natural Instruction

### Input Examples Existing Technology

| **Linguistic** | “A truck is parked if it does not move for more than 2 minutes.”  “After a truck to truck transfer, follow the receiving truck.” | CONTROLLED ENGLISH  
CPL: Peter Clarke  
ACE: N. Fuchs |
| **Hands-On**  | Interface to simulated city with controllable UAV fleet.  
A simulator of the UAV’s vision and control system | TIELT: D Aha |
| **Sensing / Acting in world** | | |
| **Actions** | Sequence of way-points instructor used to maneuver UAV to get picture into back of truck. | PLANNING ACTION LANGUAGES  
PDDL, SPARK-L, SHOP2 |
| **Gestures** | “pointing”  
Syntax specifying which simulated world objects “this truck” and “that truck” the teacher indicated in example above | INDEXING LANGUAGES  
Xpath  
W3C – Multimodal Interaction Lang |
| **Diagrams** | Sketches (not drawn to scale) with icons depicting (a) a map of suspected terrorist safe houses and (b) surveillance goals/routes | GRAPH MODELS  
B. Chandrasekaran  
T Hammond. |
| **Lesson Structuring** | Staged curricula: Recognizing parked trucks -> Recognizing truck to truck transfers -> Behavior to get pictures of truck contents | LESSON SEQUENCING  
• SCORM, R. Farrell |

Notice many methods of Instruction humans use today can be built on this modest ladder API. For example, teaching …

- By feedback on student performance
- From examples
- By demonstration
- By giving worked solutions
- By feedback from world
- By Reasoning about failures
- By Practicing
- Etc.

### Existing technologies are sufficient for an initial API
Bootstrap Learning Components (BLCs) capture structure from one lesson for use in later learning.

- System's behavior is controlled by many interconnected BLCs.
- Bottom of ladder has initial BLCs.
- Bootstrap Learning is filling in BLCs.
- Interlingua below is the languages that BLCs are written in.
- Those 3 interlingua languages are enough for wide range of problems.
- Because Learning inputs and outputs BLCs, it can continue indefinitely.

### EARLIER LESSON
In restricted natural language instructor says: “… parked with their rears near each other”
Parsed as: \texttt{near( rear(t1), rear(t2) )}
BL does not know meaning of “Rear” but it can infer that Rear is a property of trucks, and since “Near” is spatial, it knows output of “Rear” property must be a point in space

### LATER LESSON
This ‘type’ information constrains learning of the function that computes ‘rear(truck)’

\[
\text{Rear}(t) \equiv \text{center}(t) + \frac{1}{2} \text{diameter}(t) \times \text{rotate}(180, \text{direction}(t))
\]

---

**Interlingua** (the language BLCs are written in)

<table>
<thead>
<tr>
<th>Interlingua languages</th>
<th>Examples</th>
<th>Potential Technology</th>
</tr>
</thead>
</table>
| Syntactic             | \(f(x:y)\) | UAV \texttt{isa} PhysicalObject, etc.  
RearOf(x) \equiv \texttt{fn}: PhysicalObject \Rightarrow PhysicalLocation | Word Net, Frame Net, Is-a OWL, DAML, CycL |
| Logical               | \(P \rightarrow Q\), \(R \rightarrow S\) | \text{Near-By}(a,b) \equiv \text{Dist}(\text{Loc}(a), \text{Loc}(b)) < 3\text{ meters}  
\text{if } \text{At}(\text{Safe-House, location}) \text{ then } \text{Suspicious(Truck)} | Predicate Calculus; Horn Logic Equations / Formulas |
| Procedural            | SubProcedure: Investigate Possible T2T Transfer  
(1) repeat 5 times:  
(2) move-to( hiding place); loiter(5min); move-to(truck)  
(3) if TransferInProgress then call “PhotographTransfer” | HTNs (Hierarchical Transition Nets)  
MDPs (Markov Decision Process)  
Scripting Languages  
Partial Policy – Russell |

---

Each ladder rung fills content into existing (or instantiates new) BLCs
Natural Instruction Methods

**CORE METHODS**

**Syntax Learning**
infers syntax from usage

**By Annotated Example**
examples from teacher

**By Refinement**
examples from behavior

**By Demonstration**
actions from teacher

**By Rote (From Lingual Input)**
answer explicitly stated

**Special cases of ‘By Example’**
- By Worked Example (Explanation)
- By Demonstration
- By Instantiated Plan

**Special cases of ‘By Discovery’**
- By Analogy
- By Noticing Simplifications
- By Representation Bridging

**Meta Learning**
- Adapting top-level control
- Applying methods to top-level

**By Feedback on Student Solution**
# Possible Problem Ladders

## Strategy Games
- Civilization: FreeCiv, Call 2 Power
- War Games: MadRTS

## Operator Training
- Remotely operated vehicles
- Surveillance tasks
- Situation Assessment

## Training Simulators
- Commercial and military training simulators

## Design
- Floor plan design on CAD tool
  e.g. design floor plan from cust. reqs.

## Diagnosis
- Diagnose and repair failures
  at a nuclear power plant.

## Planning/Scheduling
- Applying sequences of image correction algorithms
  to obtain optimal telescope images

## E.G. Instruct humans and machines to do:
Employ multi-level reactive strategies needed to win game

### Execute complex tasks (fly UAV)
Identify fraud, suspicious behavior, etc.

### The Army course of action planning system used in RKF

### IDEAL LADDER PROPERTIES

#### LADDERS MUST BE CHEAP TO BUILD
- Require limited background knowledge (small delta)
- Leverages existing simulators, and training materials
- Easy to providing “SAME” ladder to human.

#### LADDERS MUST PROVIDE COMPLEX INSTRUCTION
- Large increment in human performance after teaching
- Multiple layers of (sub-)concepts / (sub-)procedures.
- Requires relational knowledge & representation shifts

### Cyber domains where perception problem is easier

### “Natural” tasks (currently taught to humans)

Performers compete to provide most ladder for least cost (reusing their assets)
Bootstrap Learning Components (BLCs)

Interlingua languages

<table>
<thead>
<tr>
<th>Syntactic</th>
<th>Logical</th>
<th>Procedural</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{fn}:x \Rightarrow y$</td>
<td>$\text{P} \Rightarrow \text{Q}$</td>
<td>SubProcedure: Investi gate Possible T2T Transfer</td>
</tr>
<tr>
<td>UAV isa PhysicalObject, etc. RearOf(x) = \text{fn}: \text{PhysicalObj} \Rightarrow \text{PhysicalLocation}$</td>
<td>Near-By(a,b) $\equiv$ Dist(Loc(a),Loc(b))&lt;3 meters if Then Suspicious(Truck)</td>
<td>(1) repeat 5 times:</td>
</tr>
<tr>
<td>$\text{fn}$ step $\Rightarrow$ min</td>
<td></td>
<td>(2) move-to(hiding place); loiter(5min); move-to(truck)</td>
</tr>
<tr>
<td>Hard-coded greedy path planning algorithm</td>
<td></td>
<td>(3) if TransferInProgress then call “PhotographTransfer”</td>
</tr>
</tbody>
</table>

Potential Technology

Word Net, Frame Net, Is-a OWL, DAML, CycL, SUMO

Predicate Calculus; Horn Logic Equations / Formulas

HTNs (Hierarchical Transition Nets) MDPs (Markov Decision Process) Scripting Languages Partial Policy – Russell

Each ladder rung fills content into existing (or instantiates new ones) using the Interlingua
Approach Recap

1. Defined Ladder API
   - Isolates learning from problem specifics
   - Simplifies perception/action

2. Build Ladders.
   Each ladder encapsulates all instructor & world interactions needed for a single curriculum

3. Defined Interlingua
   BLCs are Input and Output of bootstrap learning processes

4. Learning Processes
   Input: Ladder API (#1)
   output: BLCs (#3)

   PROBLEM:
   Output BLCs are far too complex for existing domain-independent ML

   Why can BL solve what today’s ML cannot?
   BL does not use tons of data to discover structure,
   BL captures that structure from the provided instruction.

Domain Independent ML can only work by leveraging Natural Instruction
Natural Instruction Makes Complex Bootstrap Learning Feasible

Three exponential reductions in complexity

#1 *Structure of ladder* decomposes large search space into smaller spaces

#2 Structure of BLC constrains what needs to be learned

Why didn’t anybody else think of this?

#3 Within a single rung *rich inputs* constrain learning

“Ladder Rungs” reflect divide-and-conquer learning lessons that simplify search space
Bootstrap Learning Processes (identified in seed study)

### Inputs (ladder API)

<table>
<thead>
<tr>
<th>Inputs</th>
<th>LEARNING ALGORITHMS</th>
<th>Output (interlingua)</th>
</tr>
</thead>
<tbody>
<tr>
<td>in</td>
<td>Syntax Learning</td>
<td>out</td>
</tr>
<tr>
<td>in</td>
<td>Annotated Examples</td>
<td>out</td>
</tr>
<tr>
<td>in</td>
<td>By Refinement</td>
<td>out</td>
</tr>
<tr>
<td>in</td>
<td>Explanation Driven</td>
<td>out</td>
</tr>
<tr>
<td>in</td>
<td>By Watching</td>
<td>out</td>
</tr>
</tbody>
</table>

Each type of student-instructor interaction is handled by at least one Learning Process. Above is a list of several.

### Related Technologies

<table>
<thead>
<tr>
<th>NLP, CFG learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOIL, FOCL, ILP, ...</td>
</tr>
<tr>
<td>FOCL, MLNs, ...</td>
</tr>
<tr>
<td>EBL</td>
</tr>
<tr>
<td>ABL, HMM, POMDP</td>
</tr>
</tbody>
</table>

### Example Bootstrap Learning Process

The **Syntax Learning** BL process "listens" input modalities, and tries to infer new terms, new relations, and their arguments. E.g. Learns that "rear" applies to physical objects, and returns 3-D location.

```
fn: truck ⇒ location
```

"Rear"

### Example Bootstrap Learning Process

The **Annotated Examples** BL process only operates when the instructor uses annotated examples. It is related to existing example based induction but gestures and linguistic hints both add powerful constraint

"This rear near ..."

---

Everyone must specialize. Traditional ML specializes by domain. Bootstrap Learning specializes to particular Natural Instruction methods.
Curriculum for T2T example (knowledge learned on 8 rungs)

(1) Syntax Learning – “waypoints”
“hover for 15 min at the house then return to base”
Waypoint ≡ < x, y, z, minutes >

(2) Refinement: HandsOn – “duration”
In Path Planner BLC
StepDuration(wayPt) ≡
MaxSpeed * distance(wayPt) + minutes(wayPt)

(3) Syntax Learning – “rear of truck”
“A T2T occurs when two trucks park with their rear of this truck near rear of that truck…”
T2T(world) ≡ parked(T₁) ∧ Parked(T₂) ∧
near( rear(T₁), rear(T₂) )

rear ≡ fn x ⇒ location
x = truck | vehicle | PhysicalObject

(4) Annotated Examples – “rear”
“Rear”

rear(t) ≡ center(t) + ½ diameter(t)
• rotate( orientation(t), 180 )

(5) Annotated Examples – “T2T”
“T2T”

“This rear near that rear”
“This rear not near…”
T2T(world) ≡ parked(T₁) ∧ Parked(T₂) ∧
distance( rear(T₁), rear(T₂) ) < 10feet

(6) By showing – “flight path”
“if T2T then photograph like this”
Define PhotoT2T(truck, loc): . . .
SearchNear(loc); Find(truck);
Hover At( x, y, z, .5 min )
RecordImage. title="inside truck"

(7) Annotated Examples – “good image inside truck”
(use same algorithm as step 5)

(8) Refinement: GoalDriven – “generalizing hover location”
“…line up with direction of truck…”
Partial understanding:

“Ladder provides the scaffolding – statistical learning fills in the content”
THE BL METRIC

Machine Improvement after instruction as a fraction of human improvement after instruction

\[ \%p = \frac{M_{\text{After}} - M_{\text{Before}}}{H_{\text{After}} - H_{\text{Before}}} \]

Human Improvement

\[ \%s = \frac{\text{Rungs solved (not skipped)}}{\text{Total Number Of Rungs}} \]

Fraction of ladder rungs solved by the machine

- Measure performance only on the top rung
- Failures on low rung may cascade up ladder, so we allow BL to skip (look up answer) for failed rungs.
- Thus performance is measured as a function of fraction of rungs solved by learning.
- When multiplexing system across multiple ladders we always use minimum score obtained.

BL metric is relative humans perf. (\%p) and assigns partial credit (\%s)
Go / No-Go Tests

Program Claims

#1 BL solves wide range problems with zero reconfiguration

#2 BL’s learning compares with human learning

Phase 1 – new algorithms & end-to-end test

- Blue teams tested on their own DARPA approved ladder.
- Red team verifies each ladder utilizes:
  - $\geq 3$ modalities,
  - $\geq 2$ learning processes,
  - $\geq 3$ ladder rungs.

Phase 2 – human comparisons & multiple ladders

- Must attain $X\%s$ & $Y\%p$ of graduate performance multiplexed across the $3$ diversity ladders.
- Must attain $X\%s$ & $Y\%p$ on hidden human-comparison ladder

Phase 3 – Program Success Tests

- Must attain $X\%s$ & $Y\%p$ multiplexed across all $5$ diversity ladders.
- Must attain $X\%s$ & $Y\%p$ on new hidden human-comparison ladder

Generality tested by multiple ladders. Head-to-head human comparisons on hidden ladders
Protocol For Human Comparison

Natural Instruction Methods
- Syntax Learning
- By Annotated Example
- By Refinement
- By Demonstration
- By Rote (From Lingual Input)

LEARNING ALGORITHMS
Blue teams build algorithms to learn from each instruction method, and provide background components.

TEST PROBLEMS
Red teams propose & DARPA approves
DEVELOP CURRICULA
Red teams use Natural Instruction Methods to encode teaching curriculum for each problem

Ladder API specifies decoding methods for providing the “same” instructions to both human and machine.

HEAD TO HEAD COMPARISON
Red team provides “same” curricula to both humans and BL systems and measures performance improvement for each

Humans & machines use same instructional material and same tests

Next Slides ⇒ How are they compared?
Human Testing Details

All student instruction is provided directly from the computer (using the same curriculum ladder given to BL).

The top rung of each ladder has a problem generator and scoring function. E.g. “how many cities did you build in 50 turns.”

Students are tested with ZERO instruction, and tested again after learning from the ladder in order to establish improvement.

All Go/No-go thresholds must be achieved with high confidence (P > 95%). Since each student’s performance is independent we use single-tailed t-test.

20 to 40 test subjects will suffice of achieve this confidence level.

Protocol delivers a good “apples to apples” comparison
Program Deliverables

- An electronic student
  - Very Reusable learning components (because of framework)
  - Much stronger forms of learning (driven by instruction)
  - Compares to human learning performance

- Datasets drive new ‘Instruction-Based-Learning’ community
  A test harness that, for the first time, allows individual researchers to develop and test new Natural-Instruction based bootstrap learning processes.

- A domain achievement
  Trainable military technology for transition

The byproducts of the program are as useful as the program itself
Summary

This program creates an “Electronic Student” with general-purpose, indefinitely-bootstrappable learning. How?

1. Instructor provides complex structures that statistical ML could never learn

2. Bootstrap learning exponentially simplifies learning in 3 ways

3. Learning is isolated from problem specifics so it cannot depend on them. (this is the only way to get learning that can bootstrap toward any task)

4. Learning is specialized to Natural Instruction type not problem type

BAA expected in fall of 2006

Release planned after RFI workshop.
Advantages of the BL Program Structure

**Program structure is efficient for research progress**
- Learning teams are **provided data** (from many domains) in a clean consistent format.
- Learning teams are the ones that **get to define that format**.
- Complex input from the world has been **abstracted** in order to facilitate algorithms.

**Fertile ground for novel research**
- Each NI method will be provided with a **novel combination of inputs**. (imagine a sequence of actions, plus specification of current goal/sub-goal, plus instructor gesturing at relevant world features at each step.)
- Any algorithm built to take advantage of these novel inputs will be **breaking new ground** since that combination of inputs will not have been available to others.

**Datasets are specifically designed to drive publishable research**
- Claims of an algorithm’s generality are supported because the test ladders are intentionally drawn from **multiple disparate problem domains**.
- Curricula packaged to included world simulators, relevant background knowledge, and a structured tree of problem generators. Since each ladder is a self contained complete testing environment, they facilitate very **rapid development and testing** of new algorithms.
- The BL program aspires to provide datasets to drive research on instructable learning in much the same way that the **Irvine repository** drove supervised induction in the 1980s.
Ideal NI Method & Learning Algorithm

**Natural** – Method is an abstraction of a plausible interaction between human instructor & student.

**Practical** – Method would be an effective method for ‘programming’ new behaviors into computing systems.

**Robust** – Learning algorithm would handle missing/noisy inputs, as well as “haphazard” instruction

**Ubiquitous** – Method can be used extensively across a diverse range of domains

**Efficient** – Method is sensitive to the implied instructor time needed for its application

**Encodable** – Method’s interaction can be encoded as into Ladder with relative ease.

blaylock@cyc.com
Ideal Performer Attributes

Learning teams will contribute a set of NI Methods:

- Ideal proposals justify their instruction methods’ ubiquity and practicality in instructing computing systems.
- Ideal proposers have significant prior research on algorithms related to their proposed methods.
- Ideal proposals contain non-obvious algorithmic ideas about how to tractably integrate multiple sources of constraint provided to each NI method in Bootstrap Learning.
- Ideal proposals contain approaches that are robust to missing/noisy inputs.
- Ideal proposals explain how bootstrapping can be repeated and how NI methods integrate with other methods.
- Ideal proposals explain (when appropriate) how shifts in representation, and shifts in learning bias can occur.
Example Application: \textit{Field Training} new UAV behavior

**SCENARIO**

- Intel officer suspects truck-to-truck (T2T) transfers are used to get bomb materials into green zone.
- Officer field-trains his unit’s UAV fleet to opportunistically report on T2T transfers seen during its other activities.

**Instruction Method: using Controlled English**

A truck-to-truck (T2T) transfer occurs when trucks park with their rears near each other.

**Instruction Method: Annotated Examples**

See, this truck rear is near that truck rear.

This truck rear is not near that truck rear.

**Instruction Method: Explained Student Performance**

UAV finds many trucks in parking lots

the trucks must have people nearby.

**Instruction Method: By Demonstration using direct control**

If there is a T2T transfer, then take good images of the inside of the truck bay, like this.

**Lesson omitted for brevity: teaching “good image”**

UAV practices taking images based on demonstration and def of “good image”

**Instruction Method: By Practice**

Obtain as many T2T transfer images as possible while expending no more than 5% of time resources on task.

UAV adaptively updates its T2T trigger strategies to stay with the 5%

Without BL, \textit{each} new behavior needs a handmade software update
Approach Overview

PROGRAM OBJECTIVE
Creating the “Electronic Student”:

1. Ladder API formalizes interaction between student and instructor+environment

2. Provide multiple training ladders to force generality of Bootstrap Learning

3. Develop bootstrapping component “interlingua” as input/output of learning

4. Build learning processes that learn all parts of bootstrapping interlingua from Natural Instruction encoded in Ladder API

- Drive Domain-independence by testing on multiple unknown ladders

- Far more ambitious about what is learned because bootstrapping provides scaffolding

Enabling insight: separate the learning algorithm from the problem domain(s)