Learning to Perform Actions from Demonstrations with Sequential Modeling

Tuan Do
Doctoral Thesis Defense
Brandeis University
Advisor: Prof. James Pustejovsky

June 20, 2017

Motivations

- Development toward domestic robots
- Fully automated robots are far from able to perform tasks in unfamiliar environments or novel circumstances.
- Robots with behavioral robustness can learn from a broad range of experiences by operating in a dynamic human environment.
- Artificial intelligence (AI) systems that:
  - learn adaptively
  - communicate with humans on different levels of abstraction and by different modalities
  - learn new actions by mimicking human companions

Communication with Computers

- Communication with Computer (CwC) is a DARPA initiative.
- CwC takes into account multiple modalities:
  - vision
  - spoken language
  - gestures
- CwC focuses on developing technology for assembling complex ideas from basic ideas given language and context.

Learning from Demonstration

- What is Learning from Demonstration (LfD)?
  - Teach agents the concept of an action so that they can perform it in a new context
  - Is vital to next generations of adaptable AIs
- Related to Learning from Interaction (LfI), i.e., learning new concepts through communication.
- Focus of LfD
  - Learning of action skills: Is difficult because of their *temporal-spatial* dynamics.
  - Learning from videos: vision and linguitic inputs.

Action learning

- Action learning from videos: traditionally as a classification task, e.g. classify *running*, *sitting*, *eating*, and *playing sport*.
- We need to move toward more fine-grained treatment of action representation:
  - human-object interactions
  - complex activities that are combination of primitive actions
- Learning from readily available and large video dataset is desirable but very difficult.
- Examples from Movie description dataset (MPII):
  - His bike slides underneath the vehicle.
  - Someone slides across a white limo's hood.
  - The car slides into a turn.

Language of action

- Language of actions and motions: represented in verbs and adjuncts.
- manner-oriented vs path-oriented
  - The ball rolled across the room.
  - The ball crossed the room rolling.
- *manner* aspect of action: intrinsic movement gradient of an object over time.
- *path* aspect of action: relative change of position of an object w.r.t other objects.

Qualitative reasoning

- Qualitative reasoning (QR):
  - Is originally the logical reasoning strategy that humans employ to cope with an infinite amount of data.
  - Is strongly associated with natural languages.
  - Applies the same principle in designing decision-making machines.
- Qualitative spatial reasoning (QSR):
  - Is reasoning by discretizing spatial information.
  - Is basic for language of action.
  - Allows robots to process inputs, plan actions and navigate in spaces.

Problems

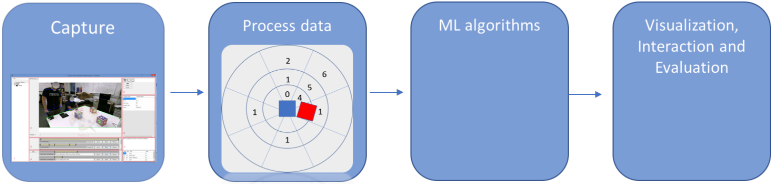
- Action performance from real demonstrations: where we learn actions from real captured demonstrations.
- Action performance from synthetic demonstrations: learn from a parallel corpus of instructions and corresponding sequences of actions as video captures.
- *Action recognizer**: a preliminary study to discover an appropriate action representation.

Problem description

- Learning to perform action skills: Slide Closer, Slide Next To, Slide Away, Slide Past, and Slide Around.
- They have fairly different action representations.
  - **closer to**: close-ended, soft ending.
  - **away from**: open-ended, soft ending.
  - **past**: open-ended, soft ending.
  - **next to**: close-ended, hard ending.
  - **around**: open-ended, soft ending.
- Individual action performance is evaluated by human judgment on 2-D demonstrations and by automatic evaluators.
- Comprehensive performance of all actions is evaluated by 3-D visualization.

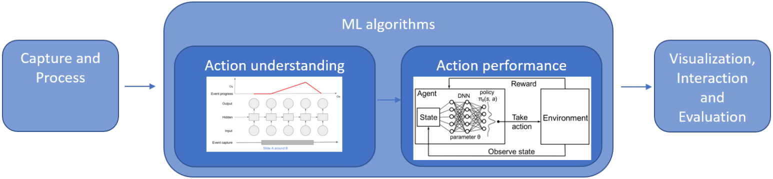
Capture, annotation, and processing

- Event capture and annotation tool (ECAT) that captures human interaction with objects using Kinect sensors.
- Feature extraction and representation through Qualitative spatial reasoning (QSR).



Machine learning algorithms

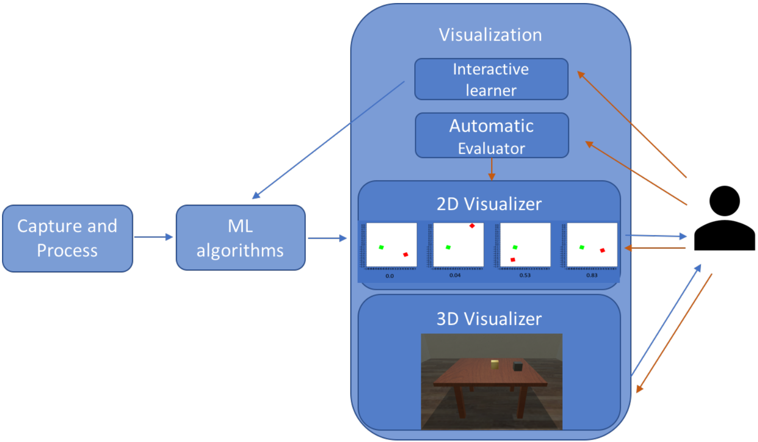
- *Action understanding* models: sequential models that feed features in frame-to-frame manner.
- *Action performance* models: sequential models that produce step-by-step actions.
  - Heuristic search algorithms on action space.
  - Reinforcement learning algorithms (RL): policy gradient algorithms
  - Hybrid between search and policy gradient algorithms.



Machine learning algorithms (cont.)

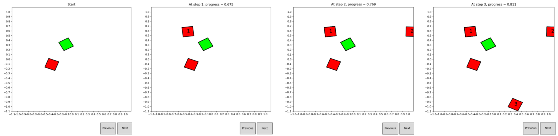
- *Action understanding*: A progress function to guide movement of actions
  - Learned by a Recurrent Neural Network (RNN) with a Long-Short Term Memory (LSTM) cell.
  - To extract a reward signal from experts' behavior.
  - Produces a value from 0 to 1.
- *Action performance*: search and policy gradient algorithms in RL setup.
  - Search algorithms: best-first search and beam-search.
  - Policy gradient algorithms: REINFORCE or ACTOR-CRITIC.
  - Searching space:
    - Continuous space: action policy is modeled as Gaussian function of state → *quantitative feature space*.
    - Discrete space: the search space is divided into regions, and actions are moves between adjacent regions → *qualitative feature space*.

Visualization, interaction and evaluation

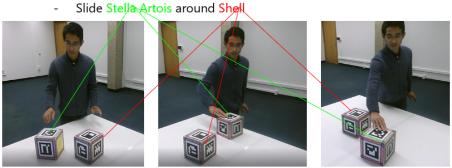


Evaluation - 2-D visualizer

- Offline mode: Generate 2-D scenes into video files (.mp4)
- Interactive mode: allows feedback from users to update action models immediately.
  - *Feedback loop*: Users can decide to correct a bad action.
  - Users can save the updated model back to files.



Experimental Design - Slide Around

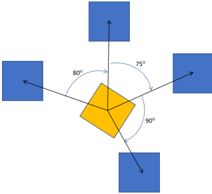


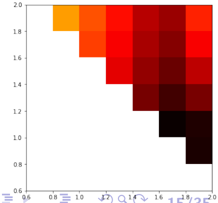
- Captures of 20 demonstrations of clockwise movement, 20 of counter-clockwise movement by 2 performers.
- Annotation is semi-automatically:
    - Performers and objects are tracked.
    - Action spans (beginning and end frames) and a description of the action are annotated in a separate session.
- Captures are sliced into equal-length chunks to be fed into LSTM to train the progress learner.

Evaluation

- Human 2-D evaluation: Hired annotators are shown demonstrations on 2-D, and asked to give a grade between 0 and 10.

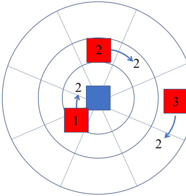
- Automatic evaluation: calculate the covering angle of the moving object around the static object
    - *Covering angle* is absolute of the sum of this value for each step.
    - Score will be either 0, 1, or 0.5, based on two threshold values.
    - By calculate correlation with human judgment, the best value combination of thresholds are $alpha_1 = 1.1$ and $alpha_2 = 1.7$.





Results

- Search algorithms produce bad results.
- Hybrid ACTOR-CRITIC gives a nice and interpretable solution: "Always go to the right", i.e., clockwise movement.
- Use of the learned action policy: greedy selection.
- Policy gradient algorithms work on discretized space, but not on continuous space.



  - Evaluation score (best possible score):
    - Human: 6.7/10
    - Automatic: 0.7/1
  - Disadvantages: computational overhead, difficulty in model formalization, need to be retrained if the action model is updated.

Feedback loop

- Purpose: methods to improve the progress function incrementally to improve search algorithms.
- *Cold feedback*: a binary value given by users indicating whether a demonstration is good or bad.
- *Hot feedback*: using the interactive mode of the 2-D simulator to correct a bad action step.

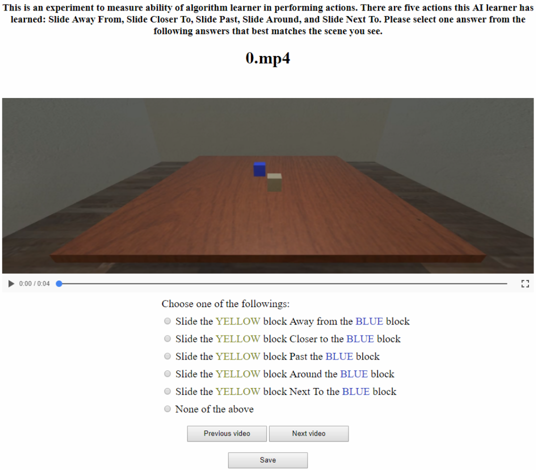
Progress function		Search algorithm			
		Greedy		One-step beam search	
		Continuous	Discrete	Continuous	Discrete
Original model		0.19	0.15	0.16	0.20
Cold feedback	Human-Feedback	0.40	0.20	0.58	0.42
	Auto-Feedback	0.38	0.20	**0.60**	**0.65**
Hot-Feedback		**0.5**	**0.37**	0.44	0.30

Mini-conclusions

- Feedback loop could be used to improve the baseline model, with different advantages.
    - *Cold-feedback* methods has an advantage when running with beam-search algorithms.
    - *Hot-feedback* methods, improving the progress function on each demonstration step-by-step, favor the best-first search algorithm.
- Perhaps, a simple search algorithm coupled with interactive improvement methods is the best way to go.

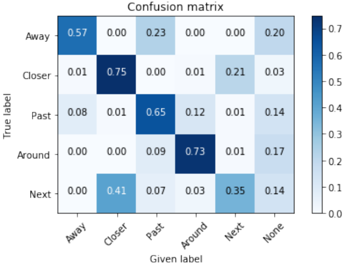
Evaluation - 3-D visualizer

- Modify from Voxeme Simulator (VoxSim), our lab's internally developed environment for generating animated scenes in real time
- The Point of View (POV) in the 3-D visualizer is similar to the POV in the captured environment.



Evaluation by 3-D simulators

- 150 demonstrations are generated (30 for each action), using the search algorithm on continuous space.

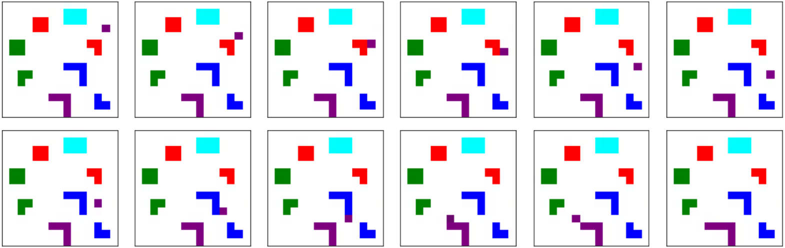


- Fleiss' Kappa value of annotator agreement gives 0.508, which is considered "Moderate agreement".
- Perhaps, we need a further step for AI learners to distinguish between different actions.

Problem description

- Purpose: given textual inputs describing actions (*instructions*), plan sequence of action steps on a grounded visual environment.
- Difference: natural language inputs; visual fields with non-trivial shapes.
- Generate synthetic demonstrations in (*maze traversal space*), posed to annotators to solicit textual descriptions.
- Training data: A parallel corpus of instructions and corresponding sequences of actions as video captures.
- In total, there are 300 videos, 200 are used for training/validating (4 annotations each), and 100 are used for testing (1 annotation each).

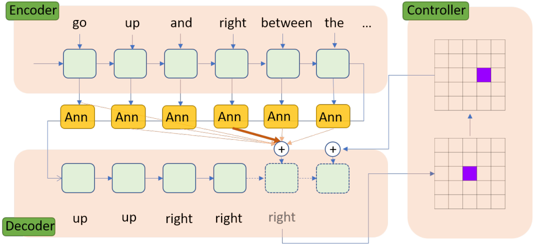
An example



- Input: move the purple square down on the right side of the blue L and red L then move the purple square left between red L and purple L and it ends at the left side of the purple L.
- Output: down, down, down, down, down, down, left, left, down, down, left, left, left, left, left, left, down, left

Learning models

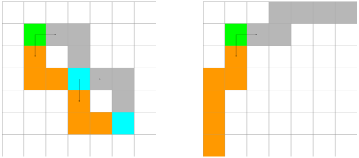
- We can use Neural Machine Translation!
- Visual grounding is incorporated as a controller model, paired with Attention Encoder-decoder.



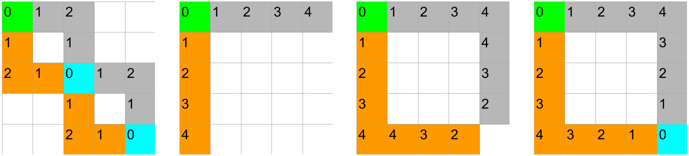
- Evaluations:
  - Internal evaluation: perplexity is a measure of objective that is used in gradient optimization method.
  - External evaluation: using NEIGHBOR score

Evaluation examples

- The internal evaluation is agnostic to *convergent* versus *divergent* paths.



- NEIGHBOR scores of the following cases: 1.07, 2.2, 2.53, 2



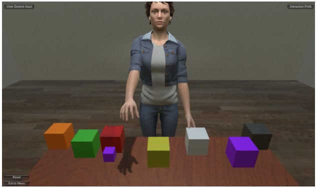
Experiments and results

- Improvement of the model with visual grounding vs non-visual grounding.

	NEIGHBOR		Avg. length	
	Eval	Test	Eval	Test
Without visual grounding	4.13	4.01	10.62	8.94
With visual grounding	**3.17**	**3.65**	**11.33**	**9.30**
Ref.	0	0	18.24	17.89

- Possible further improvements:
  - Augment the training data by shifting the cells in each puzzle while keeping the instruction and commands intact.
  - Encode our knowledge of shapes into learning models.
  - Multiple attentions: attention from both text instruction and visual environment.
  - What will happen if the instruction is incorrect?

Learning from Interaction (ongoing)



- Use of simple communicative means in 3-D CwC interface to improve learned action models.
  - *Cold feedback* can be mapped to positive and negative acknowledgments.
  - *Hot feedback* can be mapped to pointing act.
- Next step: Currently, we need some demonstrations to bootstrap the model; ongoing work on exclusively interactive learning.

Learning from Interaction (ongoing)

- Life-long learning from demonstration entails LfI with linguistic inputs.
- They provide invaluable information that is not readily available from demonstrations.

HUMAN: Let me teach you **Slide object A around object B**?
ROBOT: OK. You want to teach me an action that takes two objects A and B. Is that correct?
HUMAN: That is correct.
ROBOT: Ok. Show me a demonstration.
HUMAN: Here it is. [*Slide a red block around a green block*]
ROBOT: I saw a red block and a green block, do you mean A is the red block and B is the green block?
HUMAN: That is correct.
ROBOT: [*Calculate movements of the two objects*]. The green block did not move. Can we set the green block as the origin of the planning space?
HUMAN: Sure.
ROBOT: [*Transform geometrical space to have the origin at the green block*] Do we need the action trajectory or just the ending state?
HUMAN: The trajectory.
ROBOT: [*If trajectory then use a sequential model, else use a non-sequential model*] Does the intrinsic object model of the red block matter?
HUMAN: Not really.
ROBOT: [*Remove object model of the red block*] I have learned an action model. Let me perform it to you. This is **Slide the green block around the red block**. [*Performs the action*]
HUMAN: That is correct!

Learning from Interaction (LfI) (ongoing)

- Adding more biases to the AI learning models step-by-step.
- Prerequisite: Robots are already equipped with some spatial and action concepts.
- We can also update and correct the learned models:

HUMAN: [*After a while*] Can you slide the yellow block around the red block?
ROBOT: Sure [*Performs the action*].
HUMAN: I mean the reverse direction.
ROBOT: I do not know that. Show me a demonstration.
HUMAN: Here it is. [*Slide in the reverse direction*]
ROBOT: [*Calculate the new trajectory in the old model*] The new trajectory does not match my previous action model. Do you want to update the previous model or to separate into two models?
HUMAN: Ok, separate two action models. Name your old model **Slide object A around object B clockwise** and the new model **Slide object A around object B counter-clockwise**.
ROBOT: Ok. Two models are created.

Extensions to the methodology

- Path- versus manner- aspect of actions: to distinct between "rolling a bottle" and "sliding a bottle".
- Symbolic versus feature-based learning: we should only consider LfD as one modality for teaching action skills to AI agents in a holistic approach.
  - Teach trajectory action skills from demonstrations.
  - Teach more complex skills by communicative means.
- Virtual reality LfD: VR can be used as a shared collaborative environment between humans and machines.

Summary

- Contribute a feasibility study for a methodology to teach AI agents action skills through multimodal communicative interfaces.
- Examine different perspectives of the ML problem of planing actions given textual instructions on a visually grounded environment.
- Survey different machine learning methods for *action understanding* and *action performance*.
- Examine the compatibility of classical structural machine learning theory and modern deep learning methods.

Conclusions

- Machine learning methods, such as RL and Seq2Seq, can be used to teach skills to robotic agents by demonstrating actions to them.
- We can leverage on both advantages of classical ML theory and modern DL methods by encoding human bias into machine learning models (under the name of Qualitative spatial reasoning).
- Learning can be aided by interactive communication between machine learners and human teachers.

Conclusions

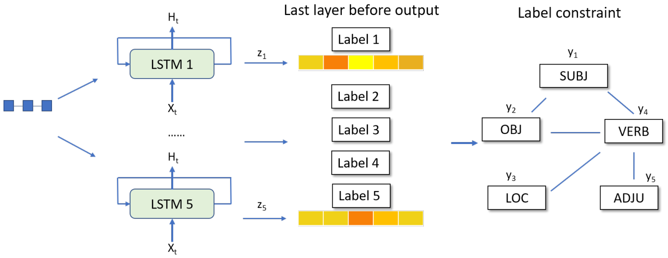
- Thank you
- Questions?

Action recognizer

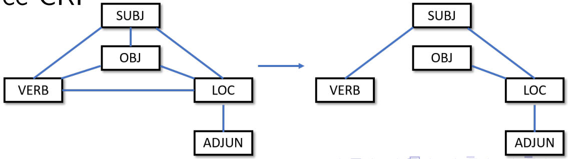
- Distinguish fine-grained action types: be used to classify different action types that combine manner-and path- aspects of motions.
- *The performer pushes A toward B* → (The performer, A, B, Push, Toward).
- Distinguish among action verbs *push, pull, slide, and roll*, along with three spatial adjuncts *toward, away from, and past*; and causative-inchoative altenation (*The performer slides A* vs *A slides*).
- Experiments with quantitative vs qualitative feature sets.

Learning model

- LSTM-CRF



- CRF → Tree-CRF



Results and mini-conclusion

- Captured sessions are split for 5-fold cross-validation, i.e., 24 sessions for training and 6 for testing on each fold. A prediction is correct if all slots are correct. Performances of different models are reported in the following tables:

Model	Precision
Baseline	6%
Quant-LSTM	39%
Quant-LSTM-CRF	48%
Qual-LSTM-CRF	**60%**

Label	Precision
Subject	93%
Object	90%
Locative	80%
Verb	83%
Preposition	82%

- Gives us the semantic treatment of actions which will become foundational to the learning to perform framework in the next chapter.
- Qualitative feature representation of actions benefits classification results → qualitative models to reenact actions.