UNIVERSITY OF LATVIA
FACULTY OF COMPUTING

Tools for Learning to Act Systematically

Guntis V. Strazds

guntis_vilnis.strazds@lu.lv

November 9, 2022

Supervisor: Prof. Guntis Bārzdiņš, Dr.sc.comp

Tools for Learning to Act Systematically

Big Picture: Interests and Goals

- **Generalizable Problem Solving**
  - Autonomous agents - goal attainment with a variety of goals and environment configurations
  - Learn from experience - from interactions with simulated environments
  - Ideally: explainable and adjustable behavior
- **Need to Publish**
  - Goal: 2 Papers this year...

UNIVERSITY OF LATVIA

Generalizable Problem Solving

Reactive, Systematically Compositional Sequential Decision Making

- **Reactive** (partially observable world)
- **Systematically Compositional**
  - Intelligently recombine from a repertoire of skills
- **Sequential Decision Making**
  - Act: (plan / choose an action)
  - Observe: see what happens
  - Adjust: react / re-plan accordingly

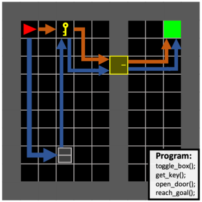
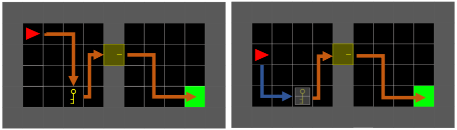


Figure 1: A motivating example.

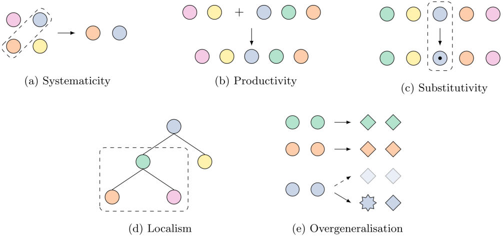


(a) DoorKey (b) BoxKey

From **GALOIS: Boosting Deep Reinforcement Learning via Generalizable Logic Synthesis**[2]

Introduction / Review

Types of Generalization

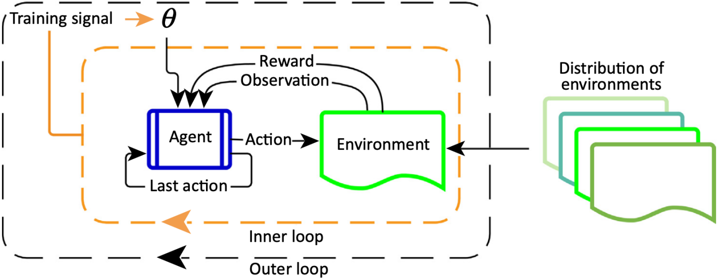


Source: Hupkes et al. 2020 - Compositionality Decomposed: How do Neural Networks Generalise? [5]

	Definition
(a) Systematicity	Recombine constituents that have not been seen together during training
(b) Productivity	Test sequences longer than ones seen during training
(c) Substitutivity	Meaning unchanged if a constituent is replaced with something equivalent
(d) Localism	The meaning of local parts are unchanged by the global context
(e) Overgeneralization	Can handle exceptions to rules and patterns?

Definitions from: `https://evjang.com/2021/12/17/lang-generalization.html`

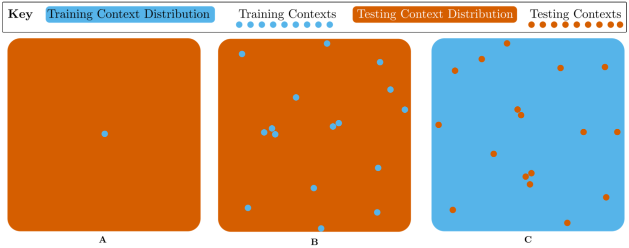
Sequential Decision Making, Generalizable across multiple task instances

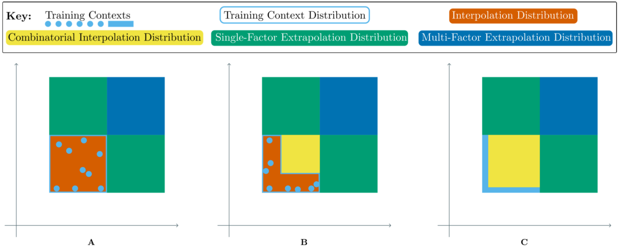


from Fig.1 of Botvinik et al. 2019 - **Reinforcement Learning, Fast and Slow** [1]

Introduction / Review

Procedurally generated tasks/envs





Kirk et al. 2021 - A Survey of Generalisation in Deep Reinforcement Learning [6]

Introduction

Approaches for Compositional Systematicity

- **RL Meta-learning**
  - Multi-task and Curriculum learning
    - Auto-curriculum: Intrinsic motiviation, "curiosity"
  - Continual ("Life-long") learning
  - In practice: Imitation Learning, offline RL
- **Hierarchical Sequential Decision Making**
  - Program Guided
  - Natural Language Instruction Following
  - LLM Guided - e.g. suggest actions, or procedure 'sketches'
- **Program induction and synthesis**
  - Procedural vs. Declarative (e.g. logic programming)
    - Planning languages; constraint satisfaction

(see also Sections 4 [Details] and 5 [Ideas])

Status Update

Context: What's new this past year?

- Trend toward super-scaling continues
- Large Language (and Language+Vision) Models (LLMs)
  - getting ever bigger & better
  - exhibit (imperfect) compositional systematicity
- Transformers are beginning to be used also for SDM
  - LL(V)Ms as generalist, do-everything models
  - but also simpler, not-so-huge generative auto-regressive models for imitation learning
    - Essentially GPT2/minGPT with very minor mods

UNIVERSITY OF LATVIA

Context

Mega-scaling vs. Inductive Biases

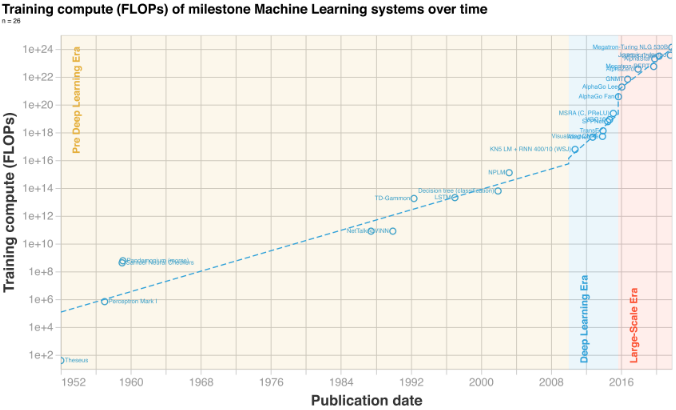


Figure from Sevilla et al. 2022 [7]

Three eras of exponential scaling

Context

Mega-scaling: Industry vs. Academia

Sutton's "bitter lesson from 70 years of AI research"[1] *Given exponentially increasing computing resources, general purpose learning and search methods end up, over a time span only slightly longer than a typical research project, outperforming knowledge-intensive, hand-crafted approaches.*

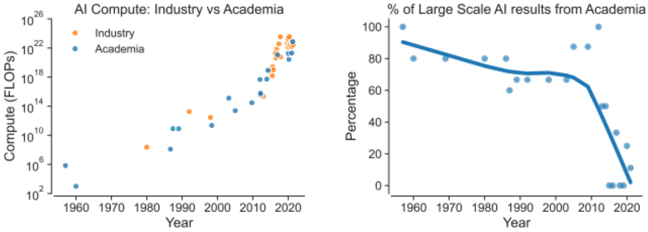


Figure from Ganguli et al. 2022 [4]

The scale of current SoA models is now beyond the reach of most academic researchers. So what can we do?

[1] `http://www.incompleteideas.net/IncIdeas/BitterLesson.html`

Status Update

My status: What have I done this past year?

- SemEval 2022 - CODWOE
- Learning, Reading
- Some coding

SemEval-2022, Task 1 CODWOE Competition

- Comparing Dictionaries and Word Embeddings
- The CODWOE shared task compares two types of semantic descriptions: dictionary glosses and word embedding representations. Are these two types of representation equivalent? Can we generate one from the other?
  - **Definition modeling track** - participants have to generate glosses from vectors. (We participated only in this one);

E. Mukans, G. Strazds, G. Barzdins

Co-located with NAACL 2022 (Annual Conference of the North American Chapter of the Association for Computational Linguistics)

Learning, Reading (quite a lot)

Sutton and Barto (2018) Intro to RL book ; Levine et al. Offline Reinforcement Learning ; parts of Distributional RL book (Bellemare et al. 2017)

	Topics
7	Newly published papers using TextWorld
11	Diffusion and Flow based models
10	NeRFs and other Neural Fields
15	OO Factoring, MoE, Causal models
23	RL/SDM Generalization: learning good representations
7	Offline and Imitation Learning
15	Planning: learning and using an env model for SDM
19	Transformers for RL/SDM, and/or more compositional or more efficient
20+	Hybrid / Neuro-symbolic / program induction
27	Cognitive Architectures and/or CogSci related

(Topics included in my curated & categorized list of "good papers")

Some Coding (not enough)

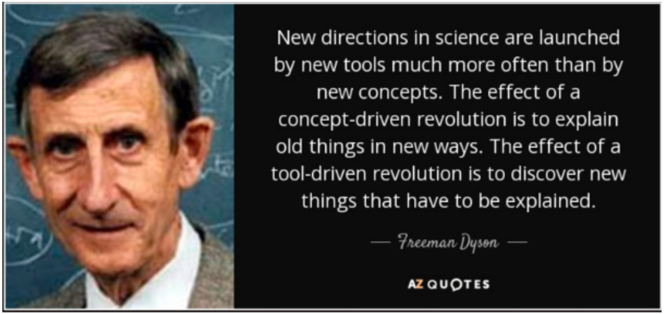
The Plan

What now?

- Goal: 2 Papers this year...
  - 1st ready to submit in January
    - Tooling for running experiments (generalization in TextWorld)
    - ... together with some initial results demonstrating usefulness
    - (Nice to Have) also integrate Minigrid
  - 2nd ready to submit in May
    - Several ideas, details in Section 5 ("Ideas") [Sorry, not really - I ran out of time while preparing this presentation]

The Plan - Paper 1 (part A): Tooling

Endorsements :-)



The Need for Open Source Software in Machine Learning

Sören Sonnenburg, Mikio L. Braun, Cheng Soon Ong, Samy Bengio, Leon Bottou, Geoffrey Holmes, Yann LeCun, Klaus-Robert Müller, Fernando Pereira, Carl Edward Rasmussen, Gunnar Rätsch, Bernhard Schölkopf, Alexander Smola, Pascal Vincent, Jason Weston, Robert Williamson; 8(81):2443–2466, 2007.

The Plan - Paper 1 (part A): Tooling

TextWorld training workbench (batteries included)

- Pytorch-lightning
- Hydra-config
- Weights and Biases (WandB) integration
- Huggingface Tokenizers, Datasets
- Huggingface pre-trained Transformer models
  - or, can train various architectures from scratch
- Other Transformer implementations / variations
  - labml - nicely documented, consistently implemented
  - FAIR xFormers – (modular, high-performance building-blocks for research on Transformer architecture variations)

The Plan - Paper 1 (part B)

Initial results (demonstrating usefulness of the platform)

Systematic investigation: disentangling the factors that make TextWorld games challenging. Which factors are responsible for how much of the difficulty?

- Parsing Convoluted Natural Language descriptions
- Large action space, variable length commands
- Partial Observability
- Combinatorial variation in env layouts and goals

Do any of the new (compositional or RL-specialized) Transformer variants work significantly better than others?

- Do some of the difficulty factors still prove challenging?
- (Optional) How about for a pre-trained LLM?

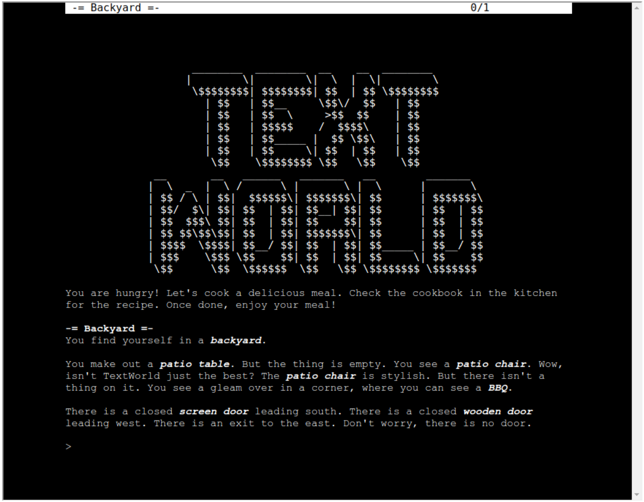
The Plan - Paper 1 (Part B)

Some candidate Transformer variants to evaluate

- Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context
- (Decision Transformer) Decision Transformer
- (Trajectory Transformer) Offline Reinforcement Learning as One Big Sequence Modeling Problem
- Online Decision Transformer
- Switch Trajectory Transformer with Distributional Value Approximation for Multi-Task Reinforcement Learning
- Unsupervised Learning of Temporal Abstractions with Slot-based Transformers
- Block-Recurrent Transformers
- R-Transformer: Recurrent Neural Network Enhanced Transformer
- Making Transformers Solve Compositional Tasks
- Coordination Among Neural Modules Through a Shared Global Workspace
- Transformers are Sample Efficient World Models
- All You Need Is Supervised Learning: From Imitation Learning to Meta-RL With Upside Down RL
- (GATO) A Generalist Agent
- Can Wikipedia Help Offline Reinforcement Learning?
- Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity
- Chain of Thought Imitation with Procedure Cloning

More Details - TextWorld

TextWorld – A Platform for Text Adventure Games



`https://www.microsoft.com/en-us/research/project/textworld/try-it/`

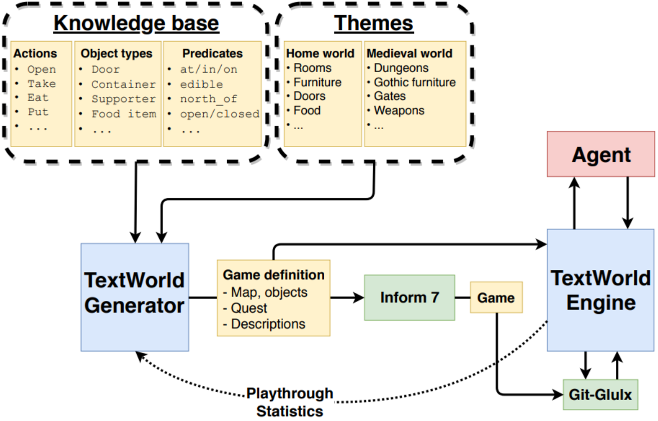
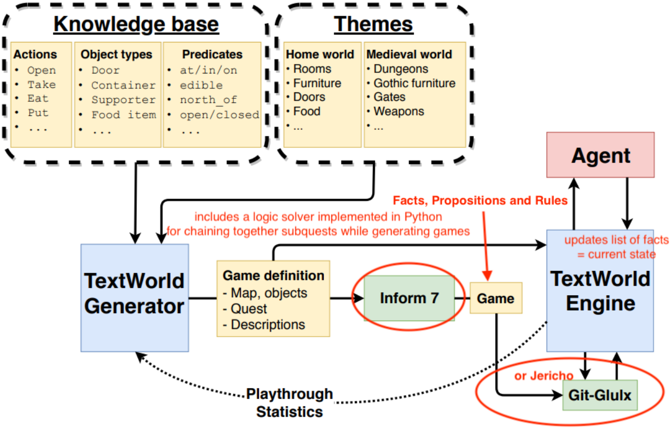


Figure from Côté et al. 2019 - **TextWorld: A Learning Environment for Text-Based Games** [3]

(click here for list of available challenges)

TextWorld Engine

TextWorld – Game Runtime



TextWorld - Challenging for RL

- Partial visibility - Player can see only what's in current room, only in opened containers
- Large and fairly complex action space
- Objects can be ON or IN other objects, or carried
- Object state attributes (based on object type):
  - open / closed / locked
  - cut / chopped / sliced / diced
  - roasted / baked / fried / ...
- Hierarchical type system (supports multi-inheritance)
- Verb + direct object + instrument
  - **peel the purple potato with the knife → a peeled purple potato**
- Long referring expressions: ***the sliced roasted yellow Idaho potato***
- Limited inventory capacity

TextWorld - Natural Language Parsing

Intentionally convoluted / obfuscated textual descriptions

You arrive in a kitchen. A normal kind of place.

You can make out a fridge. Empty! What kind of nightmare TextWorld is this? As if things weren't amazing enough already, you can even see an oven. You wonder idly who left that here. Empty! What kind of nightmare TextWorld is this? You lean against the wall, inadvertently pressing a secret button. The wall opens up to reveal a table. On the table you can see a cookbook. As if things weren't amazing enough already, you can even see a counter. The counter is vast. On the counter you can see a raw red potato. Suddenly, you bump your head on the ceiling, but it's not such a bad bump that it's going to prevent you from looking at objects and even things. Oh, great. Here's a stove. I guess it's true what they say, if you're looking for a stove, go to TextWorld. But the thing is empty.

There is an open plain door leading east. You don't like doors? Why not try going west, that entranceway is not blocked by one.

TextWorld - Simpler representations

Human-readable facts & Transformer-friendly text


```
at(P, kitchen: r)
at(counter: s, kitchen: r)
at(fridge: c, kitchen: r)
at(oven, kitchen: r)
at(stove, kitchen: r)
at(table: s, kitchen: r)
on(cookbook: o, table: s)
on(red potato: f, counter: s)
open(fridge: c)
open(plain door: d)
west_of(exit_w: e, kitchen: r)
east_of(exit_e: e, kitchen: r)
east_of(plain door: d, kitchen: r)
```

```
-= kitchen =-
IN +open fridge : nothing ;
IN +open oven : nothing ;
ON table : cookbook ;
ON counter : +raw red potato ;
ON stove : nothing ;

Exits
east +open plain door to
  unknown ;
west to livingroom ;
```

TextWorld - Internal state representation

Raw facts, with internal ids


```
at(P, r_5: r)
  at(c_0: c, r_0: r)
  at(oven_0: oven, r_0: r)
  at(s_2: s, r_1: r)
  at(s_1: s, r_0: r)
  at(s_5: s, r_5: r)
  at(s_0: s, r_0: r)
  at(s_4: s, r_3: r)
  at(s_3: s, r_4: r)
  at(stove_0: stove, r_0: r)
  base(f_0: f, ingredient_0: ingredient)
  chopped(f_0: f)
  chopped(ingredient_0: ingredient)
  closed(c_0: c)
  closed(d_0: d)
  cookable(f_0: f)
  cooked(f_0: f)
  cooking_location(r_0: r, RECIPE)
  cuttable(f_0: f)
  east_of(r_5: r, r_2: r)
  east_of(r_2: r, r_4: r)
  east_of(r_0: r, r_1: r)
  edible(f_0: f)
  edible(meal_0: meal)
```

```
free(slot_8: slot)
free(slot_2: slot)
in(f_0: f, I)
in(ingredient_0: ingredient, RECIPE)
ingredient_1(f_0: f)
link(r_1: r, d_0: d, r_0: r)
link(r_0: r, d_0: d, r_1: r)
north_of(r_0: r, r_2: r)
north_of(r_2: r, r_3: r)
on(o_0: o, s_0: s)
on(o_1: o, s_1: s)
out(meal_0: meal, RECIPE)
raw(f_0: f)
raw(ingredient_0: ingredient)
sharp(o_1: o)
south_of(r_3: r, r_2: r)
south_of(r_2: r, r_0: r)
used(slot_0: slot)
west_of(r_4: r, r_2: r)
west_of(r_1: r, r_0: r)
west_of(r_2: r, r_5: r)
```

TextWorld - Built-in Logic Engine

Propositions, Actions, Rules, Command Templates

cook/toaster/cooked/needs_cooking :: cook {f} with {toaster}
– inform7_event: [cooking the {f} with the {toaster}]
$at(P, r) & $at(toaster, r) & $in(f, I) & needs_cooking(f) &
inedible(f) -> grilled(f) & edible(f) & cooked(f)

cook/toaster/cooked/raw :: cook {f} with {toaster}
$at(P, r) & $at(toaster, r) & $in(f, I) & raw(f) -> grilled(f) & cooked(f)

dice :: dice {f} with {o} – inform7_event: [dicing the {f} with the {o}]
$in(f, I) & $in(o, I) & $sharp(o) & uncut(f) -> diced(f)

drink :: drink {f} – inform7_event: [drinking the {f}]
in(f, I) & drinkable(f) & used(slot) -> consumed(f) & free(slot)

Other Environments

MiniGrid Example

Partial observability; openable boxes; unlockable (with appropr key) doors

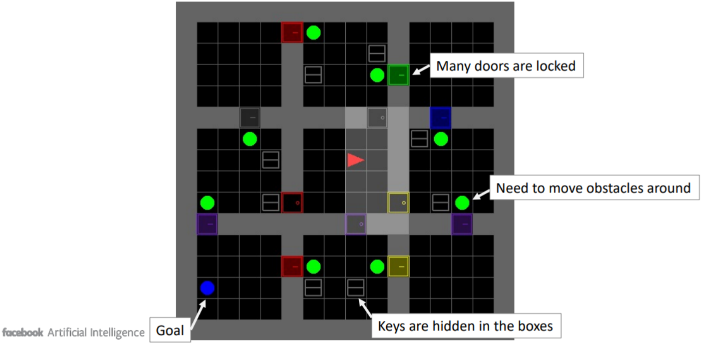


Figure reproduced from `https://https://yuandong-tian.com/ucl_dark_talk_2021.pdf`

Ideas

The Plan, Paper 2

This page not quite intentionally left blank

UNIVERSITY OF LATVIA

References I

[1] M. Botvinick, S. Ritter, J. X. Wang, Z. Kurth-Nelson, C. Blundell, and D. Hassabis. Reinforcement Learning, Fast and Slow. *Trends Cogn. Sci.*, xx, 2019.

[2] Y. Cao, Z. Li, T. Yang, H. Zhang, Y. Zheng, Y. Li, J. Hao, and Y. Liu. GALOIS: Boosting Deep Reinforcement Learning via Generalizable Logic Synthesis. may 2022.

[3] M. A. Côté, Á. Kádár, X. Yuan, B. Kybartas, T. Barnes, E. Fine, J. Moore, M. Hausknecht, L. El Asri, M. Adada, W. Tay, and A. Trischler. TextWorld: A Learning Environment for Text-Based Games. In *Commun. Comput. Inf. Sci.*, volume 1017, pages 41–75, jun 2019.

References II

[4] D. Ganguli, D. Hernandez, L. Lovitt, N. DasSarma, T. Henighan, A. Jones, N. Joseph, J. Kernion, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, D. Drain, N. Elhage, S. E. Showk, S. Fort, Z. Hatfield-Dodds, S. Johnston, S. Kravec, N. Nanda, K. Ndousse, C. Olsson, D. Amodei, D. Amodei, T. Brown, J. Kaplan, S. McCandlish, C. Olah, and J. Clark. Predictability and Surprise in Large Generative Models. *ACM Int. Conf. Proceeding Ser.*, 1:1747–1764, feb 2022.

[5] D. Hupkes, V. Dankers, M. Mul, and E. Bruni. Compositionality Decomposed: How do Neural Networks Generalise? *J. Artif. Intell. Res.*, 67:757–795, 2020.

References III

[6] R. Kirk, A. Zhang, E. Grefenstette, and T. Rocktäschel. A Survey of Generalisation in Deep Reinforcement Learning. nov 2021.

[7] J. Sevilla, L. Heim, A. Ho, T. Besiroglu, M. Hobbhahn, and P. Villalobos. Compute Trends Across Three Eras of Machine Learning. pages 1–8. Institute of Electrical and Electronics Engineers (IEEE), sep 2022.

Thank You.

(to be continued...)

guntis_vilnis.strazds@lu.lv